StorageTek™
ACSLS
AUTOMATED CARTRIDGE SYSTEM LIBRARY SOFTWARE

ADMINISTRATOR'S GUIDE

Part Number: 316120101
Version: 7.3

Automated Cartridge System Library Software

Administrator's Guide

Version: 7.3

316120101

We welcome your feedback. Please contact the Sun Learning Services Feedback System at SLSFS@Sun.com or

Sun Learning Services
Sun Microsystems, Inc.
One StorageTek Drive
Louisville, CO 80028-3256
USA

Summary of Changes

7.3 EC released document table.

EC	Date	Edition	Revision	Description
EC000249	April 2008	First	A	ACSLS 7.3 provides support for the SL3000.

Contents

Contents

Contents

Contents

Contents

Contents

Contents

List of Figures

List of Tables

Preface

■ About this Book

Automated Cartridge System Library Software (ACSLS) is Sun StorageTek's UNIX server software that controls a Sun StorageTek Automated Cartridge System (ACS). The Sun StorageTek ACS family of products consists of fully automated, tape cartridge-based data storage and retrieval systems. ACSLS supports network access to different client systems that can range from workstations to mainframes to supercomputers running on a variety of operating systems.

Audience

This Installation Guide is for the individual responsible for installing ACSLS. It is expected that you already have a working knowledge of the following:

- UNIX file and directory structure

- How to use UNIX commands and utilities for your platform

- UNIX system files

- How to do typical UNIX system administrator tasks, such as logging on as root and setting up user accesses to a UNIX application

About the Software

This guide supports ACSLS 7.3.

Conventions for Reader Usability

Conventions are used to shorten and clarify explanations and examples within this book.

Typographic

The following typographical conventions are used in this book:

- **Bold** is used to introduce new or unfamiliar terminology, or it's used in steps to indicate either an action or a decision the user has to make.

- `Letter Gothic` is used to indicate command names, filenames, and literal output by the computer.

- `Letter Gothic Bold` is used to indicate literal input to the computer.

- *`Letter Gothic Italic`* is used to indicate that you must substitute the actual value for a command parameter. In the following example, you would substitute your name for the "username" parameter.

  `Logon username`

- A bar (|) is used to separate alternative parameter values. In the example shown below either username or system name must be entered.

  `Logon username|systemname`

- Brackets [ ] are used to indicate that a command parameter is optional.

- Ellipses (...) are used to indicate that a command may be repeated multiple times.

- This guide shows all ACSLS commands in lowercase. You can, however, enter these commands in all lowercase, all uppercase, or any combination of uppercase and lowercase. Single underlines show minimum command abbreviations. For example, aud and au are valid forms of the audit command.

Keys

Single keystrokes are represented by double brackets [[ ]] surrounding the key name. For example, press [[ESC]] indicates that you should press only the escape key.

Combined keystrokes use double brackets and the plus sign (+). The double brackets surround the key name and the plus sign is used to add the second keystroke. For example, press [[ALT]]+C indicates that you should press the alternate key and the C key simultaneously.

Enter Command

The instruction to "press the <Enter> key" is omitted from most examples, definitions, and explanations in this book.

For example, if the instructions asked you to "enter" `Logon pat`, you would type in `Logon pat` *and* press <Enter>.

However, if the instructions asked you to "type" `Logon pat`, you would type in `Logon pat` and you would *not* press <Enter>.

Symbols

The following symbols are used to highlight text in this book.

WARNING: **Information necessary to keep you from damaging your hardware or software.**

CAUTION: **Information necessary to keep you from corrupting your data.**

Hint: Information that can be used to shorten or simplify your task or they may simply be used as a reminder.

Note: Information that may be of special interest to you. Notes are also used to point out exceptions to rules or procedures.

Related Documentation

ACSLS Documentation

The *ACSLS Documentation CD-ROM*, is automatically shipped with the product package and is provided in PDF format. These documents are:

- *ACSLS 7.3 Product Information*
- *ACSLS 7.3 Administrator's Guide*
- *ACSLS 7.3 Messages*
- *ACSLS 7.3 Installation Guide*

ACSLS Information on the Sun Website

In addition to the PDF collections on the *ACSLS Information CD-ROM*, the Sun website provides PDF collections for ACSLS. The URL is:

http://docs.sun.com

316120101

Overview

■ What is ACSLS?

Automated Cartridge System Library Software (ACSLS) is Sun StorageTek's server software that controls a Sun StorageTek tape library. An Automated Cartridge System (ACS) is a group of tape libraries connected through pass-thru-ports (PTPs). ACSLS accesses and manages information stored in one or more ACSs through command processing across a network. The software includes a system administration component and interfaces to client system applications, and library management facilities.

■ The acssa and acsss User IDs

To control and interact with ACSLS, you use the following user IDs:

`acssa`
> lets you enter ACSLS commands from `cmd_proc`.

`acsss`
> lets you run ACSLS utilities from the UNIX command line prompt. You can also open a cmd_proc from the acsss user ID if you prefer to work from a single user ID; see for more information.

You can log into multiple windows with both the acssa and the acsss user IDs. This lets you enter both ACSLS utilities and commands as shown in Figure 1.

Figure 1. Example Screen Display with multiple windows
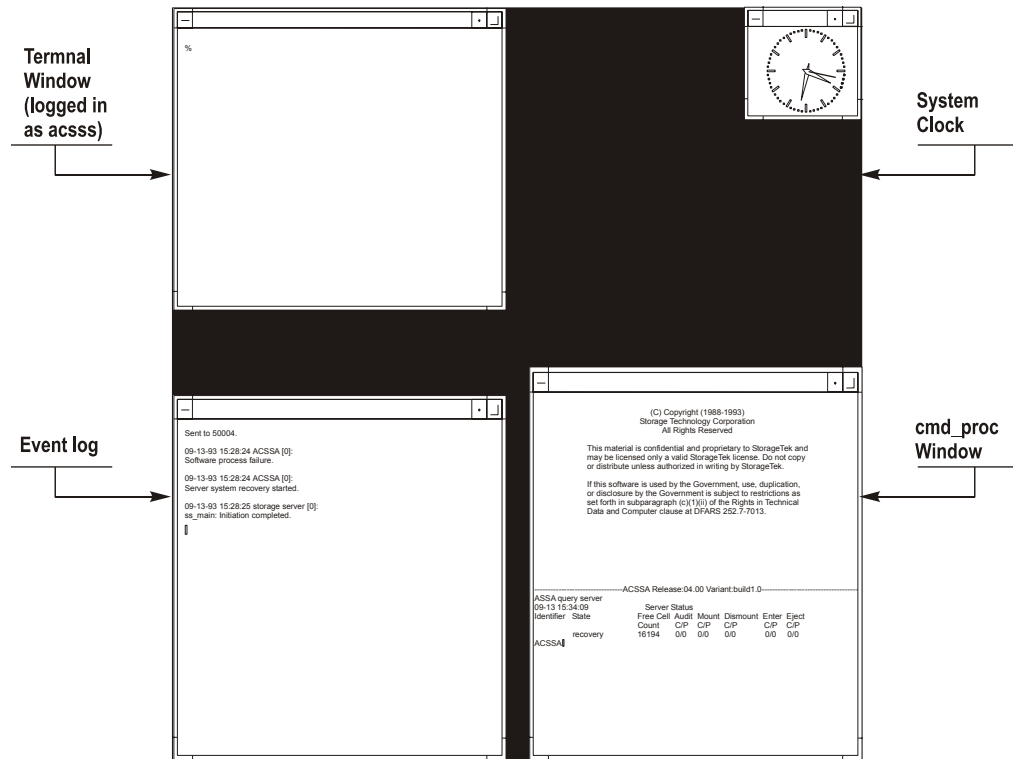
The screen display shown in Figure 1 contains the following windows:

- **A UNIX terminal window**

  logged in as the `acsss` user.

- **A "tailed" ACSLS Event Log**

  that lets you monitor ACSLS activity.

- **A** `cmd_proc` **window**

  that lets you enter ACSLS commands.

- **A system clock**

  showing the current date and time.

■ Starting ACSLS

You can start ACSLS in either of the following ways:

- Automatically at workstation boot (if the ACSLS server system startup file references the `rc.acsss` utility).

Hint: The startup file `/etc/rc` on the ACSLS server contains comments that describe how to modify this file to enable automatic startup.

- Manually by running the `rc.acsss` utility.

When you start ACSLS, the server is in recovery mode. This happens because during its initialization the database and recovery status of the library cartridges and library hardware is checked.

To manually start ACSLS, do the following:

1. Open a UNIX terminal window.

2. If you are on a remote system, to access the ACSLS server, enter the following command:

   `telnet hostname`

Where *hostname* is the UNIX hostname of the ACSLS server. Otherwise, go to Step 3.

3. Log in as `acsss`.

4. Enter the following command:

   `rc.acsss`

   Hint: You usually start ACSLS in the run state. To start ACSLS in the idle state, enter the following command:

   `rc.acsss idle`

■ Stopping ACSLS

Use this procedure to shut down ACSLS and the database. Typically, you would use this procedure before:

- Performing maintenance on the ACSLS server.

- Backing up the entire ACSLS server disk.

   Hint: Backing up the entire ACSLS server disk is not the same as backing up the ACSLS database using the `bdb.acsss` utility. See Chapter 8, "Database Administration" for procedures for backing up and restoring the database.

CAUTION: Do not use the following procedure before upgrading to a new release of ACSLS! See the appropriate chapters for specific procedures for doing an upgrade installation.

To stop ACSLS, do the following:

1. If you are not already logged in as acsss do so now.

2. Bring up a `cmd_proc`.

You can do this either on the acsss user ID or you can log in to `acssa`.

3. From the `cmd_proc`, enter the following commands to idle the server and log off from the `cmd_proc`:

    ```
    idle
    logoff
    ```

4. On the acsss user ID, bring up a UNIX terminal window, enter the following command to stop ACSLS:

    ```
    kill.acsss
    ```

    If `kill.acsss` cannot find a process ID file, the following message appears:

    `kill.acsss: PID file` (`/tmp/acsss.pid`) does not exist, library server not terminated.

    If this message appears, do the procedure in , then continue with Step 5.

5. From the UNIX terminal window, enter the following command to shut down the database:

    ```
    db_command stop
    ```

    Wait until you receive the database shutdown message.

■ Using a cmd_proc

cmd_proc Window

Figure 2 shows the `cmd_proc` window displayed when you log in as `acssa`. The `cmd_proc` window is a split screen where the top section is the message area and the bottom section is the command area. You enter ACSLS commands at the prompt.

ACSLS must be running to accept commands. If you see the message `Query: Query failed. Process failure`, wait up to a minute and re-issue the `query server` command.

Figure 2. cmd_proc Window

```
----------------------------ACSLS x.x.x----------------------------
  ACSSA> query server
  2008-01-23 15:41:42


                          Server Status
  IdentifierStateFree CellAudit Mount   DismountEnter   Eject
                  Count       C/P     C/P   C/P       C/P     C/P

         run     234         0/0     0/0   0/0       0/0     0/0
  >ACSSA
ACSSS>
```

How to Suspend and Resume a cmd_proc

If you use "Manually Starting a cmd_proc" on page 8, you can suspend the `cmd_proc` to perform UNIX commands, and then resume the `cmd_proc`.

Hint: Note the following:

1. You must start the `cmd_proc` manually.

2. You must use the C, K, or Bourne shell for this procedure.

3. Any in-process requests that you initiated at the `cmd_proc` will continue to completion while the `cmd_proc` is suspended.

To suspend and resume a `cmd_proc`**, do the following:**

1. While running a `cmd_proc`, press <CTRL>+Z.

2. The UNIX shell prompt appears.

   Perform whatever UNIX operations you want.

3. To resume the `cmd_proc`, enter the following UNIX command:

   `fg`

How to Exit a cmd_proc

Use this procedure to terminate an interactive `cmd_proc` session.

To terminate an interactive `cmd_proc` **session, do the following:**

1. While running a `cmd_proc`, wait until all in-process activity is complete and the ACSSS> prompt has returned.

2. To exit the `cmd_proc`, enter the following command:

   `logoff`

   The `cmd_proc` terminates.

   WARNING: **Do not "kill" a cmd_proc process or exit the session by closing the shell window. Terminating the cmd_proc without the proper "logoff" may lock up a shared memory segment which is used by other ACSLS processes.**

Manually Killing the ACSLS Process

If `kill.acsss` cannot find a process ID file, you must manually kill the ACSLS process.

Note: To verify if ACSLS is running, enter:

`psacs`

To manually kill the ACSLS process, do the following:

1. From a UNIX command prompt, enter the following command:

   `kill.acsss`

2. If the acsss_daemon and other processes remain in the process table, you can kill the daemon and its children with the command:

   `kill 'pid acsss_daemon'`

Starting a cmd_proc

You can start a `cmd_proc` from any terminal type with a minimum 24 x 80 display size. The terminal type must be defined in the workstation's `/etc/termcap` file.

After logging in, you can start additional `cmd_procs` by entering the `cmd_proc` command from a UNIX terminal window as described in "Manually Starting a cmd_proc" on page 8. Typically, you will run only one `cmd_proc`, but you can run as many as your system resources will allow.

Hint: A `cmd_proc` session runs independent of ACSLS. If you start a `cmd_proc` session without starting ACSLS, there will be no response to your commands. You may see a socket communication error in the `cmd_proc` if you attempt to run commands while ACSLS is not running. You need not re-start this `cmd_proc` session. It will respond normally once ACSLS is up and running.

Logging in as acsss from the ACSLS Server

To start a `cmd_proc` **when logged in as** `acsss` **on the ACSLS server, do the following:**

1. From the ACSLS server, open a UNIX terminal window.

2. To log in as `acsss`, enter the following command:

   `login acsss`

3. To start the `cmd_proc`, enter the following command:

   `cmd_proc`

4. Respond to the prompt by entering your terminal type.

   Example of terminal types are `sun`, `sun-cmd`, `xterm`, and `dtterm`.

5. Press [[RETURN]].

   The `cmd_proc` window appears.

Logging in Remotely as acssa

To start a `cmd_proc` **by logging in remotely as** `acssa`**, do the following:**

1. From a UNIX host on the network, open a UNIX terminal window.

2. To access the ACSLS server, enter the following command:

   `rlogin` *hostname* `-l acssa`

   Where *hostname* is the UNIX hostname of the ACSLS server.

   Example of terminal types are `sun`, `sun-cmd`, `xterm`, and `dtterm`.

3. Press [[RETURN]].

   The `cmd_proc` window appears.

Logging in as acssa from a Non-UNIX Client

Use this procedure to start an interactive command `cmd_proc` from a non-UNIX client on the network.

Hint: To use this procedure you must have a TCP/IP link to the ACSLS server. Depending on your installation, you may be directly connected to TCP/IP, or you may have to access it manually.

To start a `cmd_proc` **from a non-UNIX client:**

1. If necessary, access TCP/IP.

    See your system administrator for instructions on how to access TCP/IP at your location.

2. To access the ACSLS server, enter the following command:

    `telnet` *hostname*

    Where *hostname* is the UNIX hostname of the ACSLS server.

3. Log in as `acssa`.

4. Respond to the prompt by entering your terminal type.

    Example of terminal types are `sun`, `sun-cmd`, `xterm`, and `dtterm`.

5. Press [[RETURN]].

    The `cmd_proc` window appears.

Manually Starting a cmd_proc

Typically, you will manually start a `cmd_proc` from the `acsss` user ID if you do not want to log in as `acssa` to bring up a `cmd_proc`.

T**o manually start a** `cmd_proc`**, do the following:**

1. While logged in as `acssa` or `acsss`, open a UNIX terminal window.

2. To start the `cmd_proc`, enter the following command:

    `cmd_proc`

3. Respond to the prompt by entering your terminal type.

    Example of terminal types are `sun`, `sun-cmd`, `xterm`, and `dtterm`.

4. Press [[RETURN]].

    The `cmd_proc` window appears.

cmd_proc options

When you manually start a `cmd_proc`, you can also enter the following options:

`-q`
> suppresses the automatic `query server` command.

`-l`
> brings up `cmd_proc` in command line mode (command area only, no split screen, no message area).

cmd_proc Keyboard Shortcuts

Table 1 describes the `cmd_proc` keyboard shortcuts, which are <CTRL>+ keystroke combinations.

Table 1. cmd_proc Keyboard Shortcuts

Key Combination	Action	Notes
<CTRL>+C	Cancels the last `cmd_proc` command.	<CTRL>+C is the keyboard shortcut for the `cancel` command. See "cancel" on page 194 for more information about the `cancel` command.
<CTRL>+D	Returns to the `cmd_proc` prompt.	<CTRL>+D has no effect if the current command has completed. If the current command is processing, it completes but `cmd_proc` does not display a response message. If you have not entered the current command at the ACSSS prompt, <CTRL>+D deletes the command.
<CTRL>+H	Deletes the previous character on the command line.	On most keyboards, you can also use the [[DELETE]] or [[BACK SPACE]] key.
<CTRL>+I	Refreshes the `cmd_proc` display	This function is useful if the current `cmd_proc` display has been corrupted by noise on the communications lines.
<CTRL>+R	Refreshes the current command line.	This function is useful if the current command line display has been corrupted by noise on the communications lines.
<CTRL>+U	Deletes the current command line.	
<CTRL>+Z	Suspends `cmd_proc` and escapes to the shell environment.	Enter the C shell `fg` command to resume `cmd_proc`.

Redirecting cmd_proc Inputs and Outputs

You can use an input file to automatically enter commands when you start a `cmd_proc`. For example, the following input file verifies ACSLS by mounting and dismounting a cartridge.

```
query drive 0,0,0,0

query volume JB1400

mount JB1400 0,0,0,0

dismount JPB1400 0,0,0,0 force

logoff
```

To start an additional `cmd_proc` **and specify an input file, do the following:**

1. To start the `cmd_proc`, enter the following command:

    `cmd_proc -q < ` *filename*

You can also start a `cmd_proc`, specify an input file, and redirect the output to another file. Using input and output files lets you run a set of commands at `cmd_proc` startup and look at the results. For example, the following file shows the results of the commands run in the previous example that showed `cmd_proc` with only an input file.

```
ACSSA> query drive 0,0,0,0
1998-06-30 18:23:08
IdentifierStateStatus        Cartridge          Type
0,0,0,0online available                              9840
ACSSA> query volume JPL1400
1998-06-30 18:23:09
IdentifierStatusCurrent location
JB1400home     0,0,3,0,0
ACSSA> mount JPL1400 0,0,0,0
ACSSA> Mount: JB1400 mounted on 0,0,0,0
ACSSA> dismount JPL1400 0,0,0,0 force
ACSSA> Dismount: Forced dismount of JB1400 from 0,0,0,0
ACSSA> logoff
ACSSA>
```

To start an additional cmd_proc**, specify an input file, and redirect the output, do the following:**

1. While logged in as `acssa` or `acsss`, open a UNIX terminal window.

2. To start the cmd_proc, enter the following command:

   `cmd_proc -q <` *file1* `>` *file2*

   Where *file1* is the input file and *file2* is the file to which the output is directed.

   By default, `cmd_proc` display area messages are written to `stderr.` but you can also redirect these messages. For example:

   `cmd_proc -q <` *file1* `>` *file2* `2>>` *file2*

Idling ACSLS

Use this procedure to suspend request processing by putting ACSLS in the idle state. Typically, this procedure is used before shutting down ACSLS, but you can also use it to temporarily stop ACSLS request processing.

To idle ACSLS, do the following:

1. From a `cmd_proc`, enter the following command:

   `idle`

   ACSLS processes all current requests, rejects all new requests, and goes into the idle state.

Restarting ACSLS

Use this procedure to resume request processing by putting ACSLS in the run state. Typically, you restart ACSLS to remove it from the idle state.

To restart ACSLS, do the following:

1. From a `cmd_proc`, enter the following command:

   `start`

   ACSLS resumes request processing.

■ ACSLS Directory Structure

Table 2 shows a listing of the directories, subdirectories, and most common used files and shell scripts in ACSLS directory structure.

Two variables are used for ACSLS paths. They are:

- $ACS_HOME

  This is the home directory for the acsss user ID. It is also where the ACSLS product is installed.

  - For Solaris, $ACS_HOME is `/export/home/ACSSS` by default. However, the parent directory of $ACS_HOME is the ACSLS installation directory (by default `/export/home`), which is customer-defined during ACSLS installation on Solaris.

- $ACSDB_BKUP

  This is the directory where the ACSLS backups are saved.

Table 2. ACSLS Directory Structure

Directory	Contents
`$ACSDB_BKUP` (by default `/export/backup/`)	Database backups
`/etc/`	Miscellaneous UNIX files.
`/etc/hosts`	List of machines on the network and their addresses.
`/etc/passwd`	Basic information for each authorized user on the network.
`/etc/networks`	Information describing the TCP/IP network.
`/etc/netmasks`	Network masks used to implement IP standard subnetting.
`/etc/rc`	Command script; includes instructions to start up ACSLS automatically on system reboot.
`$ACS_HOME` (by default `/export/home/ACSSS/`)	Home directory for the `acsss` user ID. Also the ACSLS home directory. (By default, the `ACS_HOME` environment variable points to this directory.)
`$ACS_HOME/.acsss_env`	Defines the ACSLS environment variables.
`$ACS_HOME/.login`	Defines the ACSLS working environment; used by the C shell.

Table 2. ACSLS Directory Structure

Directory	Contents
`$ACS_HOME/.profile`	Defines the ACSLS working environment; used by the k or Bourne shell.
`$ACS_HOME/kill.acsss`	ACSLS shutdown script.
`$ACS_HOME/rc.acsss`	ACSLS startup script.
`$ACS_HOME/bin/` (by default `/export/home/ACSSS/bin/`)	Contains executable shell scripts and programs.
`$ACS_HOME/bin/acsss_config`	ACSLS configuration program.
`$ACS_HOME/bin/backup.sh`	Automatic database backup script.
`$ACS_HOME/bin/bdb.acsss`	Manual database backup script.
`$ACS_HOME/bin/db_command`	Database startup/shutdown script.
`$ACS_HOME/bin/db_export.sh`	Database export script.
`$ACS_HOME/bin/db_import.sh`	Database import script.
`$ACS_HOME/bin/rdb.acsss`	Database recovery script.
`$ACS_HOME/bin/volrpt.exe`	Executable to run cartridge report script.
`$ACS_HOME/config/` (by default `/export/home/ACSSS/config/`)	Contains ACSLS configuration files.
`$ACS_HOME/data/external/` (by default `/export/home/ACSSS/data/external`)	Contains customized files used in access control, mixed media, and cartridge reporting.
`$ACS_HOME/data/external/access_control/`	Contains access control sample and customized files.

Table 2. ACSLS Directory Structure

Directory	Contents
`$ACS_HOME/data/ external/ access_control/ adi.names`	Lists ADI names and user IDs of all hosts (OSLAN protocol).
`$ACS_HOME/data/ external/ access_control/ adi.names.SAMPLE`	Sample `adi.names` file; <u>not</u> actually used for controlling access.
`$ACS_HOME/data/ external/ access_control/ command.ALL.allow`	Lists users allowed to use all commands.
`$ACS_HOME/data/ external/ access_control/ command.`*command*`.allow`	Lists users allowed to use this command.
`$ACS_HOME/data/ external/ access_control/ command.`*command*`. disallow`	Lists users not allowed to use this command.
`$ACS_HOME/data/ external/ access_control/ command.SAMPLE.allow`	Sample command allow file; <u>not</u> actually used for controlling access.
`$ACS_HOME/data/ external/ access_control/ command.SAMPLE. disallow`	Sample command disallow file; <u>not</u> actually used for controlling access.
`$ACS_HOME/data/ external/ access_control/ internet.addresses`	Lists Internet addresses and user IDs of all hosts (TCP/IP protocol).
`$ACS_HOME/data/ external/ access_control/ internet.addresses. SAMPLE`	Sample internet.addresses file; <u>not</u> actually used for controlling access.

Table 2. ACSLS Directory Structure

Directory	Contents
`$ACS_HOME/data/` `external/` `access_control/` `ownership.assignments`	Information about assigning ownership to cartridges.
`$ACS_HOME/data/` `external/` `access_control/` `users.ALL.allow`	Lists users allowed to access owned cartridges with all commands.
`$ACS_HOME/data/` `external/` `access_control/` `users.ALL.disallow`	Lists users not allowed to access owned cartridges with any command.
`$ACS_HOME/data/` `external/` `access_control/` `users.`*`command`*`.allow`	Lists users allowed to access owned cartridges with this command.
`$ACS_HOME/data/` `external/` `access_control/` `users.`*`command`*`.` `disallow`	Lists users not allowed to access owned cartridges with this command.
`$ACS_HOME/data/` `external/` `access_control/` `users.SAMPLE.allow`	Sample cartridge allow file; not actually used for controlling access.
`$ACS_HOME/data/` `external/` `access_control/` `users.SAMPLE.disallow`	Sample cartridge disallow file; not actually used for controlling access.
`$ACS_HOME/data/` `external/` `fixed_cartridge`	Contains Extended Store Feature sample and customized files.
`$ACS_HOME/data/` `external/` `fixed_cartridge` `lsm_fixed_cartridge`	Extended Store Feature Control file; must be modified to enable this feature.
`$ACS_HOME/data/` `external/mixed_media/`	Contains mixed-media sample and customized files.

Table 2. ACSLS Directory Structure

Directory	Contents
`$ACS_HOME/data/ external/mixed_media/ scratch_preferences. dat`	Lists transport types and a (user-defined) prioritized list of media types to use when selecting scratch cartridges.
`$ACS_HOME/data/ external/mixed_media/ scratch_preferences. SAMPLE`	Sample scratch preferences file; <u>not</u> actually used for controlling media.
`$ACS_HOME/data/ external/volrpt/`	Contains cartridge report sample and customized scripts.
`$ACS_HOME/data/ external/volrpt/ owner_id.volrpt`	Sample custom cartridge report input file.
`$ACS_HOME/data/ internal/` (by default `/export/home/ACSSS/ data/internal/`)	ACSLS internal files. Please do not modify.
`$ACS_HOME/data/ internal/mixed_media/`	Contains ACSLS internal mixed media files. Do not modify these files.
`$ACS_HOME/data/ internal/mixed_media/ drive_types.dat`	Lists supported transport types.
`$ACS_HOME/data/ internal/mixed_media/ media_cleaning.dat`	Lists transport types and compatible cleaning cartridge types.
`$ACS_HOME/data/ internal/mixed_media/ media_compatibility. dat`	Lists transport types and compatible media types.
`$ACS_HOME/data/ internal/mixed_media/ media_types.dat`	Lists supported media types.
`$ACS_HOME/diag/`	Contains diagnostic files and shell scripts.
`$ACS_HOME/diag/bin/ pinglmu.sh`	Post-configuration tool to check connections between ACSLS server and serial-attached libraries.

Table 2. ACSLS Directory Structure

Directory	Contents
`$ACS_HOME/diag/bin/` `probe-scsi.sh`	Post-configuration tool to check connections between ACSLS server and SCSI libraries.
`$ACS_HOME/diag/bin/` `README.txt`	Describes the utilities in `/export/home/` `ACSSS/diag/bin.`
`$ACS_HOME/diag/data/` `ivp.dat`	Data file used to drive the IVP mount/dismount activities.
`$ACS_HOME/diag/ivp/` `ivp.sh`	Initial Verification Program (IVP) used for testing the mount/dismount commands.
`$ACS_HOME/diag/ivp/` `README.ivp`	Explanation of the IVP process.
`$ACS_HOME/install/`	Contains ACSLS installation scripts.
`$ACS_HOME/install/` `full_disk.sh`	Database automatic checkpoint backup script.

`$ACS_HOME/lib/`
 	
(by default
`/export/home/ACSSS/`
`lib/)`
	Contains ACSLS installed shared libraries required at runtime.

`$ACS_HOME/log/`
 	
(by default
`/export/home/ACSSS/`
`log/)`
	Contains ACSLS event log and utility event log files.

`$ACS_HOME/log/` `acsss_config.log`	Report of the library hardware configured by the `acsss_config` program.

`$ACS_HOME/log/`
`acsss_config_event.`
`log`
	Logs events for the `acsss_config` program.

`$ACS_HOME/log/` `acsss_event.log`	ACSLS event log.
`$ACS_HOME/log/` `bdb_event.log`	Logs events for the `bdb_acsss` script.
`$ACS_HOME/log/` `cron_event.log`	Logs events for the cron job that calls the `full_disk.sh` script.
`$ACS_HOME/log/` `export_event.log`	Logs events for the `db_export.sh` script.
`$ACS_HOME/log/` `full_disk.sh.log`	Logs events for the `full_disk.sh` script.

Table 2. ACSLS Directory Structure

Directory	Contents
`$ACS_HOME/log/ import_event.log`	Logs events for the `db_import.sh` script.
`$ACS_HOME/log/ install.log`	Logs events for the `install.sh` script.
`$ACS_HOME/log/ rdb_event.log`	Logs events for the `rdb.acsss` script.
`$ACS_HOME/log/ scsilh.log`	Logs events for SCSI LSM activity while in diagnostic mode.
`$ACS_HOME/log/ sd_event.log`	Logs events for the `sd_mgr.sh` script.
`$ACS_HOME/log/ volrpt.log`	Logs events for the cartridge report script.
`$BASEDIR/ACSSA/` (by default `/export/home/ACSSA/`)	`acssa` home directory.
`$BASEDIR/ascbd/acsbd` (by default `/export/home/ascbd/`)	Database home directory.
`/tmp/`	Temporary files.
`/tmp/ acsss.pid`	ACSLS process ID file; used in ACSLS startup and shutdown.

ACSLS License Key

<div style="text-align: right; font-size: 2em;">2</div>

◼ ACSLS License Key Utility

A license key limits the number of cartridges in your library, not the number of slots. Even if you add additional libraries to your configuration you do not need a new license key until you need to add additional cartridges.

Example: If you have a license key for 6000 cartridges and have purchased an additional library, you can move the 6000 cartridges between the libraries and not need a new license key. However, if you add additional cartridges, you will need a new license key.

After you install your license key, you have a 75-day trial period. Before your 75-day period expires, you must secure a permanent license key. Do not wait until this trial period expires to secure your permanent key.

Keep in mind that you have a full 75-day trial period once you have installed ACSLS. When you have 60 days left on your trial period, and if you have not entered your permanent license key, ACSLS will begin displaying warning messages notifying you of the date when your trial period will expire. You must request and enter your permanent license key before it expires.

Once the trial period key expires, ACSLS automatically terminates, and you cannot re-start ACSLS.

The `licensekey.sh` utility activates the license key that allows you to use ACSLS. To run the license key utility script, enter:

```
licensekey.sh
```

Permanent: Enables you to use ACSLS when Sun has received the initial license fee for a product or feature.

Note: After the trial period expires and a valid license key has not been entered, ACSLS automatically shuts down and will not re-start.

Obtaining a License Key

1. Access the following URL.

   http://www.sun.com/software/licensingcenter/storage.xml

2. Scroll to ACSLS and click ACSLS x.x.x Key Request.

   The ACSLS x.x.x Key Order Form displays.

3. Select your software level and platform type.

4. Specify the key type.

Note: You must furnish the following information:

• The Sun Sales Order Number.

• Your Customer Name as it appears on your Sun Software License Agreement.

• Name, telephone number, and e-mail address of the person within your company who is to receive the key.

License Key Format

License key parameters must be entered **exactly** as received from Sun.

Example:

```
licensekey.sh CUSTOMER_NAME 097531 2003057 ACSLSxxx
FWZZCH*DYZUP383Q
```

Note: If the customer name consists of more than one word (that is, it contains embedded blanks), the customer name must be enclosed in double quotation marks.

Hint: Copy and paste from your license key email to avoid typing errors.

An ACSLS license key consists of the following fields:

1. The customer name (64-character maximum, valid characters include A-Z, 1-9, *, and blank). All letters must be upper case.

   Note: If the customer name consists of more than one word (that is, it contains embedded blanks), the customer name must be enclosed in double quotation marks.

   CUSTOMER_NAME is the customer name.

2. The *REFERENCE/SITE* number, consisting of from four to six numeric characters.

   nnnnnn is the *REFERENCE/SITE* number.

3. The expiration date, expressed in seven numeric characters (in ISO standard date format).

   yyyyddd is the license key expiration date.

Hint: The license key expiration date is expressed as:

- yyyy = year expressed in four numerals, such as 2008.
- ddd = the number of days into the year counting from January 1 (the Julian day of the year), such as, 229 is the two-hundred twenty-ninth day of the year.

4. A product identification number, consisting of eight alphanumeric characters

   product_identifier is composed of the product name (ACSLS) followed by the three (3) character release ID, such as 7.0.0.

5. An encoded license key between 14 and 17 characters long.

   license_key_string is the encrypted license key option string.

License Key Validation

ACSLS validates the license key: during product initialization; at a preset time each day; before cartridges are entered; and after audits complete. ACSLS shuts down when the 75-day trial period ends or the emergency license key expires. After the trial period, the product will not initialize at system startup if you do not have a valid ACSLS license key. While the ACSLS system is up and running, warning messages are issued as you approach your licensed library capacity, or as you approach the trial period or license key expiration dates.

Your ACSLS license supports a maximum number of cartridges. When you exceed your licensed cartridge limit, you cannot enter additional cartridges. If you need to increase your library capacity, you must upgrade your ACSLS license to support the additional cartridges.

Use the `get_license_info` utility to display the number of cells (cartridges) supported by your current license key. The `get_license_info` utility also displays the number of used (allocated) cells, free cells, and total available cells in all libraries managed by ACSLS. Refer to for procedures.

When you exceed your licensed cartridge limit, you cannot enter additional cartridges.

License Key Operational Procedures

License keys are entered, updated, or replaced by invoking the license key script from the command line.

1. To initialize ACSLS with a valid license key,

    *   After configuring the database with `acsss_config`, but prior to entering `rc.acsss`, run the `licensekey.sh` script.

    *   Enter the required fields described above.

        Hint: Copy and paste from your license key email to avoid typing errors.

    *   Enter: `rc.acsss`

        ACSLS initializes.

2. To update or replace an ACSLS license key,

    *   Run `licensekey.sh` with the correct parameters. This can be done online without recycling ACSLS.

        ACSLS should accept the new license key.

3. If you experience errors with your ACSLS license key,

    *   Consult the *ACSLS Messages Manual, Version x.x.x*, and follow the instructions for the error message you received.

    *   If you are still unable to resolve the problem, see your Sun support representative for assistance.

Installing and Configuring Your Library Hardware

<div style="text-align: right">**3**</div>

The installation and configuration of your library hardware involve the following tasks:

- "Installing Connections to Your Library Hardware" on page 24

  Installing connections to library hardware involves:

  - Adding a the SCSI mchanger device driver

  - Optionally enabling Dual TCP/IP support (for the SL8500 or SL3000)

- "Using acsss_config to Configure Your Library Hardware" on page 24

  This lets you:

  - Establish library communications

  - Configure the number of libraries to be supported and if they are partitioned

  - Establish if the connection to a library is SCS/Fibre, TCP/IP, serial or SCSI/Fibre and the format to use

  - Reconfigure your library hardware

Note: Refer to the *ACSLS Product Information* guide for a list of ACSLS supported: libraries; tape drives; tape media; and transport/media compatibility.

■ Installing Connections to Your Library Hardware

Adding the SCSI mchanger Device Driver

The SCSI Media Changer (mchanger) is the SCSI/Fibre attached library device driver that communicates between ACSLS and any SCSI/Fibre attached library. An mchanger must be created for each SCSI/Fibre attached library that is connected to ACSLS.

Adding an mchanger device is part of the installation process. However, there may be times when you need to add a new SCSI library to an existing ACSLS installation. In this case, you must create a new mchanger device driver. To do this:

1. Login to the ACSLS server as `root` user and enter your password.

2. Go to the installation directory:

   `cd/export/home/ACSSS/install`

3. Make sure that the SCSI library is functional and physically attached to the ACSLS server.

4. Invoke the driver installation script:

   `./install_scsi_sol.sh`

Enabling SL8500 or SL3000 Dual TCP/IP Support

If you have installed the SL8500 or SL3000 library and want to enable dual TCP/IP support, refer to following sections: "ACSLS Dual TCP/IP Support " on page 344 or "ACSLS Dual TCP/IP " on page 364. These sections also discuss the customized routing table entries that you need to create.

■ Using acsss_config to Configure Your Library Hardware

Using `acsss_config` lets you configure or reconfigure your libary hardware.

WARNING: All ACSs must have at least one CAP. This can be a shared CAP.

• Configure your library hardware

   You must use the `acsss_config` utility to configure your library hardware for the first time. The `acsss_config` utility is menu driven and selecting option 8 lets you to perform tasks such as: configuring hardware for the first time; deleting libraries; or changing port connections to libraries.

Note: ACSLS 7.3 lets you to skip ACS numbers when you configure or reconfigure libraries (see # 6). Both `acsss_config` and Dynamic Configuration support specifying and skipping ACS numbers.

For example: After migrating from an ACS 0 of 9310s to an ACS 1 of SL8500s, you can remove ACS 0 without renumbering the drive and volume IDs in the remaining SL8500 ACS 1.

- Reconfigure your library hardware

Before using `acsss_config` to update the configuration information for all your libraries, record your current configuration. Changing the ACS numbers assigned to your current libraries would cause all of your volumes to be marked absent and all of your drive's addresses to change. Save the output from a `query lmu all` command to record your current ACS numbers and port connections.

After your initial library configuration, you can also use the `config` utility, to dynamically add or reconfigure ACSs, LSMs, or tape drives.

For more information and procedures, refer to "Dynamic Configuration (config) utilities" on page 133.

Note: The ACSLS database must be updated using `acsss_config` or `config` whenever the physical configuration of the library changes. ACSLS will not run correctly if the configuration defined in its database does not match the configuration defined in the library.

Examples:

Examples of configuration changes requiring an update to the database include:

- Adding or removing an ACS, LSM (including a SCSI-attached LSM such as the SL500), Pass-Thru Port (PTP), transport, or standby LMU

  CAUTION: After you install a new ACS or LSM, make sure that each attached library and LSM is fully (hardware) configured, powered on, and ready before you reconfigure ACSLS. Otherwise, `acsss_config` or `config` will not configure the library correctly.

- Adding or removing a port connection between the server system and the LMU

To start `acsss_config`:

1. Log in as `acsss`.

2. At the password prompt, press [[ENTER]].

3. Run the configuration utility:

```
acsss_config
```

The `ACSLS feature configuration` screen appears.

Note: The configuration utility, `acsss_config`, configures the library correctly unless all libraries, LSMs, and transports are not fully configured, powered on, and ready.

The following menu appears:

```
                    ACSLS feature configuration
Please enter the number followed by Return for your choice from the
following menu to configure product behavior in that area.

Press? followed by the Return key for help.
  ·   1: Set CSI tuning variables
  ·   2: Set event logging variables
  ·   3: Set general product behavior variables
  ·   4: Set access control variables
  ·   5: Set automatic backup parameters
  ·   6: Rebuild access control information
  ·   7: Event Notification settings
  ·   8: Define or Change Library Hardware Configuration
  ·   E: Exit
Menu choice:
```

Note: For a discussion of options 1-7, refer to .

4. Select option 8.

    The following prompts take you through setting up your configuration.

    *   **Prompt:** *Configure library communications? (y/n):*

        Enter `y` to establish communications between the ACSLS server and any libraries.

        Selecting **n** refreshes the configurations of your current libraries.

    *   **Prompt:** *Library server database exists and will be overwritten, continue? (y or n):* `y`

        This prompt only displays if library communications have been previously established and there is an existing database.

        If you enter **y** to this prompt, changes are made to the configuration information only.

5. Specify the number of ACSs to be supported.

- **Prompt:** *Number of ACSs to be supported:*

   Enter the number of ACSs (between 1 and 32) that to be supported at your site. You must answer at least 1 to this prompt. Hardware must be installed to support one or more ACSs.

   Note: A pair of L700e libraries connected by a pass-thru-port counts as one ACS.

   Note: Skip ACS numbering.

   You can configure or re-configure libraries without assigning the ACS numbers in sequence.

6. Specify each ACS number.

- **Prompt:** *Please enter the first ACS number [default: 0]:*

   Enter the number for this ACS.

   CAUTION: When you reconfigure existing ACSs, specify the same ACS numbers that are currently assigned to them.

   You are then prompted (see step 7) to define the device connections for each ACS.

7. Define the device connections (ports) for each ACS.

- **Prompt:** *Is ACS #n in a partitioned library? (y or n)*

   Enter **y** to create a partitioned library. The following prompt displays.

- **Prompt:** *Number of connections to ACS #n*

   This prompt appears for each ACS configured and establishes the number of communication ports for each ACS. The connections are:

   - SCSI connection(s)

      Note: An L1400 with a pair of L700e SCSI libraries connected by a pass-thru-port has two connections - one for each L700e.

   - Serial or TCP/IP connection between the ACSLS and the library. Two connections are recommended

      If you are connecting to multiple SL8500 libraries, a maximum of four connections is allowed.

   The ACSs must be physically connected. You must enter a decimal number between 1 and 4.

   Note: An L1400 with a pair of L700e SCSI libraries connected by a pass-thru-port has two connections - one for each L700e.

- • **Prompt:** *Device or host - ACS#n, connection #n:*

  Enter the device or host for each connection.

  CAUTION: Ensure that all connections specified for an ACS are actually connected to the same ACS.

  To specify the connection to a TCP/IP attached LMU, enter one of the following:

  - An IP address
  - A hostname
  - A fully qualified hostname

  Note: If a host name is specified, this host name must be mapped to the IP address entered into an SL8500 library or an 9330 LMU. This host name-to-IP address mapping is site-specific. Typically, this is done by the /etc/hosts file, your Domain Name Server, NIS, or NIS+

  WARNING: Make sure you specify the correct hostname or IP address. Do not connect to the wrong library.

  Sample TCP/IP device names for 9300 libraries

  ```
  Device or host - ACS #0, connection #0: hostname1
  Device or host - ACS #0, connection #1: hostname2
  Device or host - ACS #1, connection #0: fully_qualified_hostname
  Device or host - ACS #2, connection #0: 192.168.174.31
  ```

  Note: A connection to a 9330 ACS can be either serial or TCP/IP, but not both..

  Sample serial device names for 4400 or 9300 libraries

  ```
  Device or host - ACS #0, device #0: /dev/ttya
  Device or host - ACS #0, device #1: /dev/ttyb
  ```

  Fibre or SCSI-attached libraries are connected through mchanger devices.

  Sample device name for SCSI Library

  ```
  Device or host - ACS #1, connection #0: /dev/mchanger2
  ```

  An L1400 with a pair of L700e SCSI libraries connected by a pass-thru-port has two connections - one for each L700e.

  Sample device name for pair of L700e libraries

  ```
  Device or host - ACS #1, connection #0: /dev/mchanger2
  Device or host - ACS #1, connection #1: /dev/mchanger3
  ```

  Note: If you have more ACSs to define, you will be asked to specify their ACS numbers (see step 6).

8. Create or update the configuration in the ACSLS database.

- **Prompt:** *This step builds a database image of your complete library hardware configuration. Before proceeding, make sure that your library is completely configured, that all subsystems are functional and powered on. Build/Verify library configuration? (y or n):*

  Enter **y.**

  Before proceeding, make sure that your library is completely configured, that all subsystems are functional and powered on.

  This step updates the database with all configuration changes except adding or removing port connections.

- **Prompt:** *Library server database exists and will be overwritten, continue? (y or n):*

  Enter **y**. As the script builds the library configuration, it displays the following message of each panel in each LSM.

  ```
  ACS # n, LSM # nn, PANEL # nn, created
  ```

  The script also generates a library configuration report and appends it to the following file:

  ```
  /export/home/ACSSS/log/acsss_config.log
  ```

9. Optionally, configure non-TCP/IP clients on Solaris.

- **Prompt:** *Configure client system interfaces?* (y or n):

  Respond (y or n) when the system asks whether to configure client system interfaces.

  Enter n unless you are using an ICL client with OSLAN protocol.

  Entering **y** displays the following message:

  ```
  CSI SELECTION TABLE

  1) OSLAN CSI Not Selected
  2) ONC/RPC CSI Always Selected

  Do you want to change the CSI selection table (n):
  ```

  If y, the following message appears:

  ```
  Select OSLAN CSI (n):
  ```

  ONC/RPC is always selected.

  Note: `acsss_config` automatically backups the database and then terminates.

10. Run `rc.acsss` to start the server.

  You can look at the event log to verify everything is up and running.

If the `automatic startup on reboot` option was enabled during ACSLS installation, ACSLS is initiated automatically when the server system is rebooted.

11. Perform an audit of the Library

An audit updates the ACSLS database to match the actual inventory of library cartridges.

ACSLS is now ready for library operations. If you want to enter ACSLS commands, log in as `acssa`.

Setting Variables that Control ACSLS Behavior

<div style="text-align:right">

4

</div>

You must use the `acsss_config` menu to set the variables that control ACSLS behavior. With options 1-8, you will be:

- "Setting CSI Tuning Variables" on page 33

- "Setting Event Logging Variables" on page 36

- "Setting General Product Behavior Variables" on page 37

- "Setting Access Control Variables" on page 42

- "Setting Automatic Backup Variables" on page 43

- "Rebuilding Access Control Information" on page 44

- "Defining Event Notification Settings" on page 44

- "Configuring or Updating your Library Hardware" on page 46

    Note: You need to reboot your system if you have set any static variable(s). You do not have to do this for the dynamic variables.

To save or view current variables:

- Display the current settings of the variables by entering:

    dv_config -d

- Save the current settings of the variables to a file:

    dv_config -d > *filename*

- View the file:

    vi *filename*

- Print values of dynamic options:

    dv_print

■ Accessing the ACSLS Feature Configuration Menu

When you install or upgrade ACSLS, system defaults have already been set based on most user environments. However, when necessary, you can use options 1-8 to change any of these settings. Options 1 through 7 allow you to change dynamic and static variables.

Use option 8 to add and define your library hardware. Use this option if you have just installed ACSLS for the first time, or, for example, when you need to add a new library.

Select **?** to get detailed information on each prompt.

To start `acsss_config`:

1. Exit out of the CDE until you reach the CDE login.

2. Log in as `acsss`.

3. At the password prompt, press [[ENTER]].

4. Run the configuration script:

   `acsss_config`

   The `ACSLS feature configuration` screen appears.

   Note: The configuration script, `acsss_config`, will fail unless all LMUs, LSMs, and transports are fully configured, powered on, and ready.

   The following menu appears:

```
                    ACSLS feature configuration

Please enter the number followed by Return for your choice from the
following menu to configure product behavior in that area.

Press? followed by the Return key for help.

  ·   1: Set CSI tuning variables

  ·   2: Set event logging variables

  ·   3: Set general product behavior variables

  ·   4: Set access control variables

  ·   5: Set automatic backup parameters

  ·   6: Rebuild access control information

  ·   7: Event Notification settings

  ·   8: Define or Change Library Hardware Configuration

  ·   E: Exit

Menu choice:
```

■ Verifying and Changing Dynamic and Static Variables

Options 1 through 8 allow you to change the following dynamic and static variables.

Setting CSI Tuning Variables

The Client System Interface (CSI) handles communication between ACSLS and clients of other servers. It sets up how communications are handled for each client. If communication with one client is lost, the other clients are not affected and their communication continues without interruption. Multiple CSIs can run under ACSLS.

Option 1 allows you to set or change the following:

- **Prompt:** *Maximum age in seconds of pending requests in CSI request queue [172800]*

  Variable: `CSI_CONNECT_AGETIME`

  A dynamic variable, this setting determines how long ACSLS holds on to client requests to which it has not responded.

  Valid entry is: 600 to 315360000 seconds. Default is 172800 seconds.

- **Prompt:** *Number of seconds between successive retries [4]*

  Variable: `CSI_RETRY_TIMEOUT`

  The default is 4 seconds.

  A dynamic variable, this option specifies the minimum amount of time, in seconds, that the CSI should wait between attempts to establish a network connection.

  You should modify this value if timing problems occur between the CSC and CSI.

- **Prompt:** *Number of retries for the CSI before a timeout condition occurs [5].*

  Variable: `CSI_RETRY_TRIES`

  A dynamic variable, this option specifies the number of attempts the CSI should make to transmit a message. Pending messages are discarded if a connection cannot be established within the number of retries specified. Default is 5 retries.

- **Prompt:** *Changes to alter use of the TCP protocol will not take effect until the product is restarted. CSI support for RPC using the TCP protocol is enabled [TRUE].*

Variable: `CSI_TCP_RPCSERVICE`

A static option, this option specifies if the CSI is to act as a TCP RPC server. Default is true.

*   **Prompt:** *Changes to alter the use of the UDP protocol will not take effect until the product is restarted. CSI support for RPC using the UDP protocol is enabled [TRUE].*

    Variable: `CSI_UDP_RPCSERVICE`

    A static option, this option specifies if the CSI is to act as a UDP RPC server. Default is true.

*   **Prompt:** *Changes to alter use of the port mapper will not take effect until the product is restarted. Enable port mapper: (ALWAYS / NEVER / IF_DUAL_LAN_NOT_ENABLED) [IF_DUAL_LAN_NOT_ENABLED].*

    Variable: `CSI_USE_PORTMAPPER`

    Default is IF_DUAL_LAN_NOT_ENABLED. A static option, the valid options are:

    *   ALWAYS - the port mapper should always be interrogated when the CSI is unable to send a message to a client.

    *   NEVER - the port mapper should never be interrogated when the CSI is unable to send a message to a client. Select this option if clients do not support a port mapper.

    *   IF_DUAL_LAN_NOT_ENABLED - the port mapper should be interrogated only if dual LAN support has not been enabled. If dual LAN support has been enabled, then it is assumed that clients do not support a port mapper.

*   **Prompt:** *Number of ACSSURR persistent processes that should be started [1]:*

    Variable: SURROGATE_PROCESSES

    Valid entries: 0 or 1. Enter 0 if the LM Gateway is not installed.

*   **Prompt:** *TCP/IP port number that the ACSLS surrogate (ACSSURR) socket will listen on for requests from a gateway system [50300].*

    Variable: SURROGATE_PORT

    This variable applies only to the Library Management (LM) Gateway. Valid entries: 50300 - 99999.

*   **Prompt:** *Number of seconds to wait for data packets to be read on surrogate/gateway sockets:*

    Variable: SURROGATE_TIMEOUT

This variable applies only to the Library Management (LM) Gateway. Valid entries: 1 - 600.

- **Prompt:** *Number of minutes to wait before deleting a stale queue entry [5].*

  Variable: SURROGATE_QUEUE_AGE

  This variable applies only to the Library Management (LM) Gateway. Valid entries: 5 - 60.

- **Prompt:** *Automatically start CSCI at ACSLS startup (TRUE/FALSE) [FALSE]:*

  Variable: *START_CSCI_PROCESS*

  This variable determines if the CSCI process is automatically started during ACSLS startup. The default value, FALSE, means that the CSCI is not started with ACSLS. Set this variable to TRUE to start the CSCI process with ACSLS.

- **Prompt:** *Enable CSI to be used behind a firewall (user-defined inbound port) (TRUE/FALSE) [FALSE]:*

  Variable: CSI_FIREWALL_SECURE

  A dynamic variable, this setting enables or disables the firewall security definition of a single, user-defined port for incoming requests to ACSLS. Values are:

  False - select False if you do not want the ports on the ACSLS server to be restricted.

  True - select True if you want the ACSLS server to operate behind a secured firewall.

Prompt: *Port number used by the CSI to receive incoming ACSLS requests.*

Variable: CSI_INET_PORT

This variable specifies the single, user-defined port used by the CSI for incoming TCP requests from one or more clients.

This variable is used only when the firewall secure CSI is enabled by setting CSI_FIREWALL_SECURE to True.

Default: 30031

Valid entry: 1024 - 65535

Setting Event Logging Variables

Option 2 allows you to set or change the following event logs:

- **Prompt:** *Number of event log files to retain [9]:*

  Variable: `EVENT_FILE_NUMBER`

  9 is the default when ACSLS is installed or upgraded.

  A dynamic value, this option allows you to specify the number of additional event log files to retain. If enabled and the current event log file size reaches the threshold size, the log is automatically copied to another file. Once the specified number of files is reached, the data in the oldest file is overlaid.

  If you specify 0, ACSLS creates only one event log file with no additional event log files. When this file reaches its threshold size, an "Event log full" message is displayed periodically until the event log is renamed or deleted.

  Specify a number between 1 and 9 to enable and specify the number of log files to retain. When the event log reaches the threshold size, the data is moved into the file `event0.log`. The next time the threshold size is reached, the `event0.log` data is moved to `event1.log` and the older event log data is moved to `event0.log`.

Prompt: *Changes to the logging directory will not take effect until the product is restarted. What directory should logging information be in [$ACS_HOME/ log]:*

  Variable: `LOG_PATH`

  This is the directory where the log files are to be placed. By default the ACSLS log files are placed in the $ACS_HOME/log directory. An alternative path can be used if there are disk space problems in the file system that contains $ACS_HOME.

  The path must be an absolute path, that is, it must start with a / or $ACS_HOME.

  Path length cannot exceed 120 alpha characters.

Prompt: *Maximum library server event log size in Kilobytes (1000 bytes) [500].*

  Variable: `LOG_SIZE`

  This option specifies the threshold size for the Event Log in Kbytes (1,000 bytes). Default is 500.

  Valid entry: 32-2147483

- **Prompt:** *Date/time format for all logs [%Y-%m-%d %H:%M:%S]:*

  Variable: `TIME_FORMAT`

  This option specifies the format to use for printing the date and time information in the event and trace logs. The default format is %Y-%m-%d%H:%M:%S.

  Note: This format does not apply to the backup dates presented for a point-in-time restore.

Setting General Product Behavior Variables

Option 3 allows you to set or change the following:

- **Prompt:** *Enable Library Volume Statistics Gathering: (ON / OFF) [OFF]: ?*

  Variable: `LIB_VOL_STATS`

  This option specifies if the Library Volume Statistics (LVSTATS) information should be collected from the following operations: enter; eject; mount; dismount; and audit. Logging these statistics can take considerable disk space. Default is OFF.

- **Prompt:** *Number of acsss_stats log files to retain [9]:*

  Variable: `VOL_STATS_FILE_NUM`

  A dynamic value, this option allows you to specify file size and number of rollover files for the volume statistics log (acsss_stats.log). Valid entry: 0-9. Default is 9.

  9 is the default when ACSLS is installed or upgraded.

  A dynamic value, this option allows you to specify the number of additional `acsss_stats` log files to retain. If enabled and the current `acsss_stats` log file size reaches the threshold size, the log is automatically copied to another file. Once the specified number of files is reached, the data in the oldest file is overlaid.

  If you specify 0, ACSLS creates only one event log file with no additional event log files. When this file reaches its threshold size, an "`acsss_stats` log full" message is displayed periodically until the `acsss_stats` log is renamed or deleted.

  Specify a number between 1 and 9 to enable and specify the number of log files to retain. When the `acsss_stats` log reaches the threshold size, the data is moved to the file `vol_stats0.log`. The next time the threshold size is reached, the `vol_stats0.log` data is moved to `vol_stats1.log` and the older event log data is moved to `vol_stats0.log`.

- **Prompt:** *Maximum library server access_stats log size in Kbytes (1000 bytes) [500].*

  Variable: `VOL_STATS_FILE_SIZE`

  This option specifies the threshold size for the `accss_stats` log in Kbytes (1000 bytes). Default is 500.

  Valid entry: 32-2147483

- **Prompt:** *Select cleaning cartridge ordering method [VOLID_SORT]*

  Variable: `UNIFORM_CLEAN_USE`

  Valid options are:

  - VOLID_SORT - orders the cleaning cartridges by the volume identifier. This uses up a cleaning cartridge before moving to the next one. If you select this option, ACSLS uses and returns cleaning cartridges in the same order as in previous releases of the ACSLS software. This is the default value when ACSLS is installed or updated.

  - LEAST_USED - orders the cleaning cartridges by usage. If you select this option, ACSLS sorts the volume list in reverse order of usage and returns the cartridges with the fewest usages first. This spreads out usage of cleaning cartridges uniformly.

  - MOST_CAPACITY - orders the cleaning cartridges by number of uses left. If you select this option, ACSLS sorts the volume list based on the number of uses left on the cleaning cartridge and returns the cartridges with the most uses left first. This uses up all cleaning cartridges close to the same time.

- **Prompt:** *Enable Transport Auto Clean Option (TRUE/FALSE) [TRUE]:*

  Variable: `AUTO_CLEAN`

  Valid options are:

  - TRUE - when a transport requires cleaning, ACSLS automatically cleans the transport before the next mount.

    Note: The drive cleaning operation in a SCSI or fibre-attached library is managed by the library firmware, not by ACSLS. Be sure to enable drive cleaning from the LSM control panel. Otherwise, you can periodically mount a cleaning cartridge from ACSLS by manual command as needed.

  - FALSE - does not enable automatic cleaning

- **Prompt:** *Library Server startup state (RUN/IDLE) [RUN]:*

  Variable: `AUTO_START`

  This option specifies the initial ACSLS state after recovery completes during server software initiation. Valid options are:

  - RUN - user requests are processed immediately after recovery completes.

  - IDLE - prevents user requests from being immediately processed after recovery completes.

    Setting this option to IDLE could be useful if you need to vary device states or perform some operational activity before you allow server access to your users.

- **Prompt:** *Changes to the number of mount processes ACSLS supports will not take effect until the product is restarted. Number of mount processes [2]:*

  Variable: `MAX_ACSMT`

  Valid entry is 1 to 5.

  Sun StorageTek recommends that you accept the default value at initial configuration, then change the value as required. Increasing this value may improve performance.

- **Prompt:** *Enable QUERY persistent processes (TRUE/FALSE) [TRUE]:*

  Variable: `ENABLE_ACSQY`

  This options allows you to execute queries as one or more persistent processes.

  Sun StorageTek recommends that you accept the default value at initial configuration, then change the value as required. Increasing this value may improve performance. Valid options are:

  - TRUE - enables persistent query processes.

    CAUTION: With a minimum configuration system, higher than 10 query persistent processes could use up system resources and not allow ACSLS to start up.

  - FALSE - disables persistent query processes. Queries are invoked as request processes.

- **Prompt:** *Changes to the number of query processes ACSLS supports will not take effect until the product is restarted. Number of processes processes [2]:*

  Variable: `MAX_ACSQY`

This option specifies the number of persistent query processes to create, only if you enabled it in the above step. Valid numbers are 1 to 5.

Sun StorageTek recommends that you accept the default value at initial configuration, then change the value as required. For large configurations (more than eight LSMs), this value may need to be increased to improve performance.

CAUTION: If this number is set too high, ACSLS may not be able to start up properly. Either lower the number or increase the maximum allowable process per user.

- **Prompt:** *Changes to the maximum number of ACSLS processes will not take effect until the product is restarted. Number of ACSLS processes [40]:*

  Variable: `MAX_ACS_PROCESSES`

  Valid numbers are: 32 to 100.

  A transient process is used to satisfy all requests except: mount; dismount; lock; unlock; clear_lock; and query_lock. The default, 40 processes, works for all ACSLS processes except extremely large configurations.

  CAUTION: It is recommended you contact Software Support for advice before changing this value.

- **Prompt:** *Enable automatic enter tracing facility (TRUE/FALSE) [FALSE]:*

  Variable: `TRACE_ENTER`

  This option specifies if the results of automatic enter operations are written in the event log. Valid options are:

  - TRUE - enables messages to be written to the event log at the end of each automatic enter operation. This is the only method for determining why volumes were not entered during an automatic enter operation.

  - FALSE - disables this feature and helps minimize the number of messages written to the event log.

- **Prompt:** *Enable volume tracing facility (TRUE/FALSE) [FALSE]:*

  Variable: `TRACE_VOLUME`

  This option specifies if unsolicited messages are displayed when adding/deleting a volume(s) in the database. Valid options are:

  - TRUE - enables unsolicited messages to be displayed whenever a volume is added or deleted in the database. Operations that can generate this messages are: audit; mount; dismount; enter; eject; recover; and vary.

  - FALSE - disables this feature and helps minimize the number of messages written to the event log.

- **Prompt:** *Number of days to retain volumes that are identified as ABSENT or EJECTED in the database. [5]:*

  Variable: `ABSENT_VOLUME_RETENTION_PERIOD`

  A dynamic variable, it sets the number of days (0-999) to keep absent or ejected cartridges in the database. Unless you reenter the volume, ACSLS automatically deletes the volume from the database when the retention period expires.

  If you never want the volume to be deleted, enter 999. To delete the volume immediately, enter 0.

  CAUTION: Setting a high retention period can cause a large number of absent or ejected volumes to be retained. This uses more space in the database and causes it to fill up faster.

- **Prompt:** *Report ABSENT or EJECTED volumes to ACSAPI clients. [FALSE]:*

  Variable: `ENABLE_STATUS_VOLUME_ABSENT`

  Valid options are:

  - TRUE - reports absent and ejected volumes to ACSAPI clients.

  - FALSE - absent and ejected volumes are not reported to ACSAPI clients (as occurred in previous versions of ACSLS).

- **Prompt:** *Status of MISSING volumes reported to ACSAPI clients: TRUE=MISSING. FALSE=IN TRANSIT [FALSE]:*

  Variable: `ENABLE_STATUS_VOLUME_MISSING`

  This option sets the status code to be returned to ACSAPI clients for missing volumes. Valid options are:

  - TRUE - reports missing cartridges to ACSAPI clients.

  - FALSE - missing volumes are not reported to ACSAPI clients (as occurred in previous versions of ACSLS).

- **Prompt:** *Support alphanumeric volume ranges for commands and utilities. Alphanumeric ranges will include all valid vol_ids, in ASCII collating sequence. Note - Changes to alphanumeric volume range support will not take effect until the product is restarted. (TRUE/FALSE) [FALSE].*

  Variable: ALPHANUM_VOL_RANGES

  Specifies whether alphanumeric volume range support is active for commands and utilities. Valid options are:

  - FALSE (default) - when specifying volume range, the first characters of a *vol_id* range must be the same and the last variable characters must be numeric. Example: AAA000-AAA999

- • TRUE - Any valid *vol_id* for the range can be specified, as long as the alphanumeric volume range includes all valid volume IDs. Valid volume ranges consist of any combination of numbers (0-9), letters (A-Z), dollar sign ($), pound sign (#), and spaces (leading and trailing).

  Example: A1Z27BC-G$123R

- • **Prompt:** *During eject processing, send an intermediate response containing ejected volumes when the CAP is filled, rather than waiting for MAX_ID volumes to be ejected. (TRUE/FALSE) [FALSE].*

  Variable: EJECT_RESPONSE_ON_CAP_FULL

  This option specifies whether the eject process should send an intermediate response when a CAP is filled, or wait for MAX_ID volumes to be ejected. On configurations with CAPs larger than MAX_ID cells, this does not affect any behavior, as an intermediate response is still generated when MAX_ID volumes are ejected. This setting will have no effect on cmd_proc, and cmd_proc will always receive a response when the CAP has been filled.

  WARNING: Changing the value of this variable will affect ACSAPI clients that expect intermediate eject responses to always contain MAX_ID volumes.

Setting Access Control Variables

Option 4 allows you to set or change the following access control variables:

- • **Prompt:** *Changes to command access control will not take effect until the product is restarted. Access control is active for commands. (TRUE/FALSE) [FALSE].*

  Variable: AC_CMD_ACCESS

  Specifies if access control is active for commands. This allows control over which ACSLS command each network client is allowed to execute.

  The default, false, disables access control.

  Note: You must stop, then restart ACSLS for this change to take effect.

- • **Prompt:** *Default access for commands (ACCESS/NOACCESS) [ACCESS].*

  Variable: AC_CMD_DEFAULT

  Specifies the default access to commands if there is no relevant access control lists.

  The default, ACCESS, allows command access unless otherwise denied through configuration information provided in the access control files.

- **Prompt:** *Changes to volume access control will not take effect until the product is restarted. Access control is active for volumes. (TRUE/FALSE) [FALSE].*

  Variable: `AC_VOL_ACCESS`

  Specifies if access control is active for volumes. This allows control over which ACSLS clients can access specific volumes. You must restart ACSLS for this option to take effect.

  The default, false, disables access control for volumes.

  Note: **You must stop, then restart ACSLS for this change to take effect.**

- **Prompt:** *Default access for volumes (ACCESS/NOACCESS) [NOACCESS].*

  Variable: `AC_VOL_DEFAULT`

  Specifies the default access to volumes if there is no relevant access control lists.

  The default, NOACCESS, disallows volume access unless otherwise granted through configuration information provided in the access control files.

- **Prompt:** *Messages will be logged when access to commands or volumes is denied (TRUE/FALSE) [FALSE].*

  Variable: `AC_LOG_ACCESS`

  Specifies if a message is logged in the event log when access to a command or volume is denied. The command or volume id is logged, as well as the access id and host id if this option is enabled.

Setting Automatic Backup Variables

Option 5 allows you to set or change the following automatic backup variables:

- **Prompt:** *Would you like to modify the automatic backup settings? (y or n)*

  Variable: None. The automatic backup settings are saved in crontab. Please use acsss_config to modify these settings.

  Valid options are:

  - Y - If yes, you are asked if you want to back up everyday. If "yes", then you must enter the time in HH:MM format.

    If no, you must select the day(s) of the week to back up. Then you must enter the time in HH:MM format.

- N - you go to the next question regarding the retention period of SDM-managed database backup files.

- **Prompt:** *Set the number of days for the retention period of SDM-managed database backup files. See Help for explanation. [5].*

  Variable: RETENTION_PERIOD

  The option sets how many days DSM tracks the data on your storage cartridges before the cartridge is designated as not eligible for reuse. Valid number of days: 4-30

  The retention period is the period of time for which you retain backups. Transaction log files are *retained* with each backup, increasing the space needed in the backup partition. When choosing a retention period, consider the following conditions specific to your site:

  - The size and activity of your library

  - The number of automatic and manual backups performed in a given period of time

    Notes:

    When you run a manual backup with the bdb.acsss utility, an automatic backup to local disk is also done, thereby increasing the number of backups in the backup partition.

    The retention period should be greater than the interval between backups. For example, if you have a retention period of four days, you should take automated or manual backups at least three times a week, with no more than three days between backups.

  - The period of time for which you would like to retain backups and transaction log files, keeping in mind that long retention periods increase the space needed in the backup partition.

Rebuilding Access Control Information

Option 6 rebuilds your access control tables, if access control is alive and/or has been changed.

Defining Event Notification Settings

Option 7 allows you to define your event notification settings.

- **Prompt:** *Set the number of event messages sent to a client between two successive client checks by ACSLS. [100]*

Variable: `CLIENT_CHECK_MESSAGE_COUNT`

This option specifies the number of event notification messages to send a client and the minimum check interval (set with next question) before ACSLS checks to verify the registered client is still alive.

Valid entry: 10 - 1000 messages

- **Prompt:** *Enter the minimum interval between two successive client checks by ACSLS (in minutes). [30]*

  Variable: `CLIENT_CHECK_MIN_INTERVAL`

  ACSLS checks periodically if a registered client is still alive. This check is performed when the minimum interval and the event notification message count is exceeded.

  Valid entry: 1 - 600 minutes

- **Prompt:** *Enter the time a client has to respond to a client check with a check registration request (in seconds) [30]:*

  Variable: `CLIENT_CHECK_RESPONSE_TIME`

  When ACSLS checks periodically if a registered client is still alive, the client must respond with a check registration request within the specified response time. If the client does not respond within this time period, the client's Event Notification registration is cancelled.

  Valid entry: 5 - 300 seconds

- **Prompt:** *Enter the polling interval for confirming Event Notification client registration (in minutes) [60]:*

  Variable: `CLIENT_CHECK_POLLING_INTERVAL`

  ACSLS checks periodically if a registered client is still alive. At the polling interval specified, ACSLS sends a client check response to all clients who have not issued register, unregister, or check registration requests within this polling interval.

  Valid entry: 10 - 1440 minutes

WARNING: Reboot Your System If You Have Set Any Static Variable(s)

  You do not have to do this for the dynamic variables.

Configuring or Updating your Library Hardware

You must use `acsss_config` (option 8) to create your library hardware for the first time or to add new libraries or re-configure existing libraries. After your initial library configuration, you can then use the `config` utility, to dynamically add, reconfigure, and remove ACSs, LSMs and tape drives and port connections to libraries. For more information and procedures, refer to "Dynamic Configuration (config) utilities" on page 133.

See "Installing and Configuring Your Library Hardware" on page 23 for more information.

Access Control

<div style="text-align: right; font-size: 2em; font-weight: bold;">5</div>

Access Control provides ACSLS the means to secure your library resources.

- *Volume Access Control* is a mechanism that enables you to own specific volumes and to limit which other users may have access to those volumes.

- *Command Access Control* allows administrators to limit specific ACSLS commands to specific users.

Both *volume access control* and *command access control* apply to users of client applications who submit requests via the ACSAPI. *Access control* does not restrict access to administrative users who submit library requests using the *cmd_proc*.

■ Volume Access Control

Volume access control is the most widely used mode of controlled access in ACSLS. When enabled, the volumes that are owned by a specific user are accessible only to that user or to trusted other users.

The first time you configure ACSLS for volume access control follow these steps:

1. Enable volume access control in ACSLS.

2. Associate a client identity with a user name.

3. Define which other users may have access to the user's volumes.

4. Establish ownership of the volumes you wish to secure.

Enabling Volume Access Control

To enable *volume access control* in ACSLS:

1. Run the configuration utility, *acsss_config*.

    The main menu displays.

2. Select Option 4 - "Set Access Control Variables".

    Each variable is listed one at a time, and its current setting is displayed.

3. Press <Enter> to accept the current or default setting.

4. Select [TRUE] and press <Enter> when the utility displays the message "Access control is active for volumes".

5. Select one of the following when the utility displays the message "Default access for volumes ACCESS/NOACCESS"...

    • Select [ACCESS] if your goal is to *disallow* access to *specific* users and allow access to all others.

      This requires specific users to be listed in a *users.ALL.disallow* file or a specific *users.COMMAND.disallow* file. See "Defining other users that are allowed access to the user's volumes" on page 49

    • Select [NOACCESS] if your goal is to *allow* access to *specific* users and disallow access to all others.

      This requires specific users to be listed in a *users.ALL.allow* file or a specific *users.COMMAND.allow* file. See "Defining other users that are allowed access to the user's volumes" on page 49.

    Note: If you wish to log instances where access to volumes is denied, select [TRUE] in response to that prompt.

    Note: Whenever enabling or disabling volume access, you need to restart ACSLS for the change to take effect.

Associating a client identity with a user name

Not all client applications pass a user ID with its ACSLS request packets. In cases where the client is not identified by a user name, you can assign a user ID.

1. Go to the `access_control` configuration directory, `$ACS_HOME/data/external/access_control`.

2. Create a file by the name *internet.addresses* or copy the *internet.addresses.SAMPLE* file.

3. In this file, create a record for each client. Each record contains at least two fields: the client IP address followed by a corresponding user name. You can include additional fields for comments.

- Separate the fields with spaces or tabs as shown in the following example:

    654.32.10.10 ulyssis   payroll department

- You can create as many client-user associations as you have client applications.

  - Where client applications pass the user name with the ACSLS request, the *internet.addresses* file authenticates the user name with the designated IP address and denies access where both fields do not agree with the values in the request packet. Where multiple clients are hosted from a common platform, the same IP address may be included multiple times in this file and this address can be associated with as many user names as are rightly applied to that IP address.

  - Where client applications do not pass the user name with the request, the *internet.addresses* file is used to establish a user name for that client. In this case, only one user name may be associated with any client I address.

4. Register any updates to the *internet.addresses* file:

a. Run *acsss_config*

b. Select Option 6 - "Rebuild Access Control Information".

    ACSLS dynamically recognizes the change.

For SNA and OSLAN clients that do not use TCP/IP, refer to the *lu62.names* or *adi.names* file in the `access_control` directory.

Defining other users that are allowed access to the user's volumes

To grant other users access to a user's owned volumes:

1. Create a file *users.ALL.allow* or *users.ALL.disallow* in the `access_control` directory.

   You can copy the templates *users.SAMPLE.allow* or *users.SAMPLE.disallow*.

2. Add a record in the file for each owner, placing the owner's user ID at the left margin.

3. Specify affected users on the same line with each owner.

4. Separate the user names with spaces or tabs as shown in the following example.

> owner_john   user-Allie   user-andre

User names listed in the *users.allow and users.disallow* files must be unique, without regard to case. The type case of characters in the user name is ignored.

Users who are not listed on the same line with the owner are given the default (ACCESS or NOACCESS) relationship to the owner's volumes.

Note: You cannot have the same user ID in both the **users.*COMMAND*.allow and users.*COMMAND*.disallow** files for the same command or **ALL.**

5. Optionally, you can establish exceptions to the volume access policy you have defined.

Generally, users are allowed full access, or no access to volumes that are under access control. However, it is possible to allow users certain restricted access to other users' volumes.

For example, you can set a policy that allows any user to query volumes that are owned by a specific user, even though they may not mount or dismount those volumes. Exceptions can be applied to any of the commands that are affected by 'access control':

To configure volume access policy exceptions for certain commands:

- You must create a *users.COMMAND.allow* or *users.COMMAND.disallow* file (where COMMAND is replaced by the specific command you wish to grant or restrict).

- For each owner, place the owner's name at the left margin, followed by the users for whom the policy applies.

6. Register any updates to the policies you define:

   a. Run *acsss_config*

   b. Select Option 6 - "Rebuild Access Control Information".

   ACSLS dynamically recognizes the change.

Establishing ownership of the volumes you wish to secure

Volume access control applies only to volumes that have explicit ownership. Unowned volumes in the library are accessible to any user. To explicitly set volume ownership use the *cmd_proc* interface:

```
ACSSA>set owner "daffy" volume V00100-V00199
Set: owner set for volumes V00100-V00199
Set: Set completed, Success.
```

You can remove ownership in a similar fashion by using an empty string:

```
ACSSA> set owner "" volume V00100-V00199
Set: owner set for volumes V00100-V00199
```

This operation clears the ownership from all of the volumes in the range. For more information, refer to "set owner" on page 267.

Volume ownership can be set automatically by the *watch_vols* utility. For more information, refer to "watch_vols" on page 180.

Ownership policies

A policy for setting and removing ownership automatically can also be defined in ACSLS. For example, you can set a policy in which any scratch volume that is mounted becomes owned by the user who mounted it. Thereafter, the volume is secured by that user. The same policy could be enhanced to remove ownership whenever the volume is returned to scratch status. A policy could be written such that all entered volumes are assigned to a default user, or to the user who requested the enter, or if the volume was previously owned, to its prior owner. Considerable flexibility is offered with this feature.

Ownership policies are defined in the *ownership.assignments* file which resides in the access_control directory. You can set a policy in this file to assign or to un-assign ownership automatically with each *enter*, *set scratch*, or *mount scratch* operation. The *ownership.assignments* file allows you to define a default owner. Whenever a volume encounters any of these operations, its ownership can be assigned to:

*   Owner_default (the default owner)

*   Same (the previous owner)

*   Requestor (the user issuing the current request)

*   Unowned (retract ownership from the volume)

Instructions for defining ownership policies are described in detail in the *ownership.assignments* file. This file includes a complete list of commands that can be used to set volume ownership.

Verying Ownership

To verify ownership you can run *volrpt* using the *owner_id.volrpt* template.

```
cd ~acsss/data/external/volrpt
volrpt -f owner_id.volrpt
```

This produces a display of all the volumes in the library listed with their associated owner.

Volume Access Summary

The following commands are supported by Volume Access Control

Table 3. Commands used by Volume Access Control

mount	query_mount	enter	set_scratch
dismount	query_scratch	eject	set_clean
display	query_volume	lock	unlock

The following table summarizes the contexts that apply when *volume access control* is enabled.

Table 4. Volume Access is Enabled

Default access for volumes is ACCESS	**Access Allowed**	**Access Denied**
Access is by means of *cmd_proc*	X	
The specified volume is unowned	X	
The user is the owner of the volume	X	
The user is associated with the owner in *users.ALL.disallow*		X
If the user is not associated with the owner in *users.ALL.disallow*	X	
Default access for volumes is NOACCESS	**Access Allowed**	**Access Denied**
Access is by means of *cmd_proc*	X	
The specified volume is unowned	X	
The user is the owner of the volume	X	
The user is associated with the owner in *users.ALL.allow*	X	
If the user is not associated with the owner in *users.ALL.allow*		X

■ Command Access Control

Command access control allows an ACSLS administrator to restrict certain classes of commands to specific client applications or specific users across the network. Controlled access applies only to user commands that are submitted via the ACSAPI and it does not apply to local users who submit commands via *cmd_proc*.

The process to configure ACSLS for *command access control* involves four steps:

The first time you configure ACSLS for command access control follow these steps:

1. Enable command access control in ACSLS.

2. Associate a client identity with a user name.

3. Define what commands are available to which users.

Enabling Command Access Control

To enable *command access control* in ACSLS,

1. Run the configuration utility, *acsss_config*.

   The main menu displays.

2. Select Option 4 - "Set Access Control Variables".

   Each variable is listed, one at a time, and its current setting is displayed.

3. Press <Enter> to accept the current or default setting.

4. Select [TRUE] and press <Enter> when the utility displays the message "Access control is active for volumes".

5. Select one of the following:

   • Select [ACCESS] if you want is to *allow* all users access to *all* commands.

     This requires specific users to be listed in a *command.ALL.disallow* file or a specific *command.XXX.disallow* file, where:

       XXX is the command for which access control is intended

   • Select [NOACCESS] if you want to *deny user* access to commands.

     This requires specific users to be listed in a *command.ALL.allow* file or a specific *command.XXX.allow* file.

   Note: If you wish to log instances where access to commands is denied, enter "TRUE" in response to that prompt.

   Note: Whenever enabling or disabling command access, you need to restart ACSLS for the change to take effect.

Associating a client identity with a user name

Refer to the procedures under "Associating a client identity with a user name" on page 48.

Defining what commands are available to which users

This process depends upon the default behavior you have selected when you enabled *command access control*. You need to create a policy file in the $ACS_HOME/data/external/access_control directory.

- If the default behavior you defined above is [NOACCESS], you need to create a *command.ALL.allow* file that contains the user ID of each client that is to have access to all ACSLS commands. Each user ID should be listed on a separate line in the file.

  If you wish to grant only specific commands to specific users, you need to create a *command.XXX.allow* file for each command the users are allowed to execute. For example, to grant permission for specific users to enter volumes into the library, you would create a file with the name 'command.ENTER.allow' and list the ID of each qualified 'enter' user on a separate line in the file.

- If the default behavior you defined above is [ACCESS], you need to create a *command.ALL.disallow* file that contains the user ID of each client that is not to have access to all ACSLS commands. Each user ID should be listed on a separate line in the file.

Table 5. Command Access is Enabled

Default Access for Commands is NOACCESS	Access Allowed	Access Denied
The request is entered from *cmd_proc*	X	
The user_ID is listed in command.COMMAND.allow	X	
The user_ID is listed in command.ALL.allow	X	
- - All other conditions - -		X
Default Access for Commands is ACCESS	**Access Allowed**	**Access Denied**
The request is entered from *cmd_proc*	X	
The user_ID is listed in command.COMMAND.disallow		X
The user_ID is listed in command.ALL.disallow		X
- - All other conditions - -	X	

■ Logging Access Control Messages

You can set a policy to log all transactions that failed because the user was denied access. The message displays the user name and the command that was attempted.

To enable access control logging:

1. Run *acsss_config* and select Option 4 - "Set Access Control Variables"

2. Change [FALSE] to [TRUE] at the following prompt: "Messages will be logged when access to commands or volumes is denied.

3. Select Option 6 - "Rebuild access control information."

ACSLS recognizes the change and begins logging each time a command request was denied.

316120101

Library Management

<div style="text-align: right">6</div>

Managing your library(ies) may consist of the following tasks:

■ Specifiying ACS Numbers

ACSLS 7.3 lets you configure or re-configure libraries without assigning all ACS numbers in sequence.

Example:

You want to migrate from a 9310 library to a SL8500. The 9310 ACS is now ACS 0 and the SL8500 ACS is ACS 1.

Skipping ACS numbers lets you migrate all of the cartridges and drives to the SL8500 ACS and remove the 9310 ACS without having to renumber the SL8500 ACS.

If you renumbered the SL8500 ACS all volumes would be deleted, and then re-added in a subsequent audit. It would also cause all of the *drive_ids* to change.

■ Auditing the Library

An audit updates the ACSLS database to match the actual inventory of library cartridges.

How Audit Works

An audit ejects duplicate and invalid volumes. The cartridges have:

- An external label that duplicates one already scanned.

- A missing or unreadable external label and no virtual label.

- An invalid media type.

- An invalid volume ID.

ACSLS records any database changes from the audit in the event log and also displays cmd_proc messages during the audit. Audits only apply to LSM storage cells, not to tape drives or CAPs. For information about running an audit, refer to .

When to Run an Audit

You run an audit to:

- Create volume information in the database for newly configured libraries.

- Add volumes to the database when the cartridges were not entered through the CAP.

  Example: you added an LSM to your library, opened the LSM door, and manually added cartridges to the LSM.

- Resolve discrepancies between the library and the database.

  Example: run an audit if you opened an LSM door and manually removed cartridges instead of ejecting them through the CAP.

  The audit either marks the volume absent or deletes the removed volumes from the database.

  WARNING:

  You must audit an SL3000 after it is configured or re-configured to ACSLS. The library reports the potential cell locations that are not accessible to ACSLS when ACSLS audits it. Inaccessible cell locations include:

  - locations where CAPs, drives, and operator panels are installed.

  - cell locations that the robots cannot access.

- cell locations that are not licensed.

- cell locations that are not in this partition.

Audit Intervals

Audit intervals depend on several factors, including: your ACSLS configuration; whether the library is reporting cell contents from its database or by examining each cell; your library configuration; the number of database changes required; and the scope of the audit. Table 6 describes how the differences in the scope of an audit affect the audit intervals.

Table 6. How the Scope of an Audit Affects the Audit Interval

This audit takes less time	Then this audit
A diagnostic ACS/LSM	An online ACS/LSM
An ACS/LSM dedicated to the audit	An ACS/LSM processing other requests
A full panel	A (partially or completely) empty panel
A drive panel	A standard panel
An inside panel	An outside panel

In addition, consider the LSM type for the component you audit, as shown in Table 7. For the 4410, 9310, and L5500 the best time can be achieved when all the cells are full, and worst time when cells are empty.

Table 7. Average Audit Times for Supported LSMs

LSM Type	Component	Average Audit Time (Best and Worse)
4410	LSM	3 - 8 hours
9310/L5500	LSM	1.2 - 6 hours
9360	LSM	5 minutes
97xx/L700/180	LSM	1 minute
SL500	LSM	3 minutes
SL8500 or SL3000	LSM	5 minutes (if the SL8500 physical audit completes before the ACSLS audit)

Note: These times assume no other active processes. 4410 and 9310 audit times may be reduced if a high percentage of the cells audited are occupied.

■ Adding a SCSI or fibre-attached library to ACSLS

The SCSI Media Changer (mchanger) is the device driver that communicates between ACSLS and any SCSI library. An mchanger must be created for each SCSI or fibre-attached library that is connected to ACSLS.

For more information, please see "Adding the SCSI mchanger Device Driver" on page 24.

■ Using the Extended Store Feature

The following sections provide information about using the Extended Store Feature.

Note: This feature does not apply to a single LSM without a pass-thru port.

When a cartridge is mounted, its "home location" is the storage cell from which it was mounted. Normally, when a cartridge is dismounted and its home location is in a different LSM than the drive, ACSLS attempts to assign a new home location in the closest LSM (the shortest pass-thru distance from the tape drive). This is called "floating" a cartridge to a new home location.

ACSLS attempts to return a cartridge to its home location after a dismount if that home location is in an LSM that is enabled for the Extended Store feature.

Note: Events such as a label mismatch on a dismount, however, may prevent ACSLS from returning a cartridge to its home location. If such an event occurs, the cartridge is stored in the LSM from which it was dismounted.

If you have an Extended Store LSM that you use for cartridge archive, this feature helps to ensure that cartridges mounted from that LSM return to their home locations in that LSM. For example, if Extended Store LSM 3 is enabled for this feature and a cartridge from LSM 3 is mounted on a drive attached to LSM 1, after the cartridge is dismounted, ACSLS will attempt to return the cartridge to its home location in LSM 3. If LSM 3 is not enabled, ACSLS will store the cartridge in a new cell in LSM 1.

You can only enable entire LSMs for this feature. You cannot enable LSM subcomponents, such as a panel or individual cell. To enable an entire ACS for this feature, you must enable each LSM in the ACS.

Note: Enabling the Extended Store feature will increase pass-thru activity when cartridges are dismounted. This may significantly decrease library performance.

To enable an LSM for the Extended Store feature, modify the `lsm_fixed_volume` file (found in the `$ACS_HOME/data/external/ fixed_volume` directory) shown in . After you modify the file, stop and restart ACSLS to enable the specified LSMs.

Use the following conventions when you modify the sample file:

- Comment and blank lines are allowed throughout the file.

- Each LSM identifier line consists of the ACS number, followed by a comma, followed by the LSM number. No spaces or tabs are allowed between the ACS number and the LSM number.

- Each LSM identifier must be on a separate line.

- No explicit ordering of the LSM identifier lines is required.

- No explicit end-of-file delimiter is required.

Figure 3. Sample Extended Store Feature Control File

```
# This lsm_fixed_volume file must be found in the
# $ACS_HOME/data/external/fixed_volume
# directory. This is a sample lsm_fixed_volume file that may be
# edited your particular configuration.
# Comments may appear anywhere in this file, but must include a
# pound sign in the first column.
# Blanks lines are also allowed throughout the file for
# readability, and
# will be ignored.
# For all the LSM identifiers found in this file, a "best" attempt
# will be
# made to return the volume to its home location at dismount.
# A valid LSM identifier consists of the ACS number, separated by a comma,
# and followed by the LSM number. Leading or trailing blanks are
# ignored.
#      ACS,LSM
#      0,0
#      0,1
# ACS 1, LSM 0 through 3
#      ACS,LSM
#      1,0
#      1,1
#      1,2
#      1,3
```

Figure 4 on page 63 shows an example of a modified control file. In this example, the comment (#) character has been removed from the lines shown in bold to specify that LSMs 0,0 and 0,1 are enabled.

Figure 4. Modified Extended Store Feature Control File

```
# This lsm_fixed_volume file must be found in the
# $ACS_HOME/data/external/fixed_volume
# directory. This is a sample lsm_fixed_volume file that may be
# edited your particular configuration.
# Comments may appear anywhere in this file, but must include a
# pound sign in the first column.
# Blanks lines are also allowed throughout the file for
# readability, and
# will be ignored.
# For all the LSM identifiers found in this file, a "best" attempt
# will be
# made to return the volume to its home location at dismount.
# A valid LSM identifier consists of the ACS number, separated by a comma,
# and followed by the LSM number. Leading or trailing blanks are
# ignored.
#      ACS,LSM
       0,0
       0,1
# ACS 1, LSM 0 through 3
#      ACS,LSM
#      1,0
#      1,1
#      1,2
#      1,3
```

■ Managing a Mixed-Media Library

"Mixed media support" means that ACSLS supports a mixture of tape drive and media (cartridge) types in the same library. ACSLS mixed-media support prevents the LSM robot from mounting incompatible media types in a tape drive. For example, in a 9310 LSM, the robot will not mount a DD3A cartridge (helical recording) in a 9490 tape drive (parallel recording).

ACSLS mixed-media support requires Media ID labels with media characters on cartridges. For more information, refer to the *Product Information* guide.

ACSLS does **not** support mixed format recording. Because ACSLS does not have access to the data path to a tape drive, ACSLS cannot detect and prevent mixed-format recording incompatibilities.

CAUTION: **The** venter **command does not provide an option to specify the media type of the cartridge you want to enter. In a mixed-media environment, ACSLS cannot prevent tape drive/media incompatibilities for virtually entered cartridges!**

Displaying the ACSLS Mixed-Media Settings

The drives_media.sh utility displays the drive types, media types, and drive-to-media compatibility settings currently supported by ACSLS. As support is added for new drives and media, they are displayed.

To display the ACSLS Mixed-Media settings, enter:

drives_media.sh

The information is written to the screen (standard output).

To output the ACSLS Mixed-Media settings to files in the /tmp directory, enter:

drives_media.sh -f

The information will be written to three files. (If the files already exist, they will be overwritten.)

> /tmp/drive_types.txt
> /tmp/media_types.txt
> /tmp/media_compatibility.txt

Using Mixed-Media Support with 3490E Cartridges

An LSM's vision recognition system can determine the media type if the 3490E cartridge has the "E" media label on the cartridge label edge of the cartridge. The media label is a separate, single character, bar code label below the six-character cartridge label. Cartridges without the media label are considered 3480 cartridges by 1/2" cartridge standards.

Sun StorageTek recommends that when you purchase new 3490E cartridges, you order them with the "E" media label to enable ACSLS to provide automatic mixed media protection for those cartridges. Sun StorageTek strongly suggests that, if practical for your site, you apply "E" media labels to any existing 3490E cartridges without such labels to allow ACSLS to provide automatic protection for those cartridges as well. If you choose to use 3490E cartridges without "E" media labels in a mixed-media environment, you, not ACSLS, must explicitly manage those cartridges. For example:

- A 3490E cartridge without the "E" seventh media character appears as a 3480 cartridge on cartridge reports and as a `query volume` response. You must, therefore, know which 3490E cartridges are incorrectly labelled, and must explicitly mount them in a compatible drive (4490 or 9490).

- Similarly, you must explicitly manage incorrectly labelled 3490E scratch tapes as follows:

  - First, you must create a homogeneous scratch pool containing only the incorrectly labelled 3490E tapes. When you create this pool, do not specify the `overflow` option unless the common scratch pool is a homogeneous pool that contains only 3490E tapes.

- Next, you must add an entry to the scratch preferences to permit mounting 3480 scratch tapes as an alternate choice for both 4490 and 9490 drives.

- Finally, when you enter a `mount scratch` command, you must specify the pool ID of the incorrectly labelled 3490E tapes and the `ANY_MEDIA_TYPE` or `ANY_MEDIA_TYPE` keyword.

CAUTION: **If you modify the scratch preferences file as described, a `mount scratch` command for a 4490 and 9490 drive will mount a 3480 cartridge when the pool no longer contains 3490E cartridges.**

- `query mount` and `query mount scratch` commands will not return accurate media type information for incorrectly labelled 3490E cartridges. Again, you must know by volume ID which cartridges that appear as 3480 cartridges are actually 3490E cartridges.

Mixed-Media Restrictions for SCSI-Attached LSMs

Please note the following restrictions for SCSI-attached LSMs:

- Because some tape drives do not support dynamic write-protect settings by means of the control path, you must use caution when mounting cartridges using the 'read-only' option. All Sun StorageTek drives support this feature. For non-Sun StorageTek drives, you are advised to confirm that dynamic write-protection is supported. Unless the drive supports this feature, you bear the risk of losing data on cartridges that you assumed are to be mounted with the write-disable protection of a 'read-only' mount.

- ACSLS does *not* support the `venter` command for SCSI-attached LSMs with DLT tape drives.

- You cannot do virtual mounts and dismounts to DLT tape drives in SCSI-attached LSMs. Compac tape cartridges, therefore, must have external labels to allow ACSLS to manage these cartridges.

- Automatic-cleaning operations in SCSI-attached LSMs is handled by the LSM microcode and not by ACSLS. The LSM control panel provides a menu for users to enable LSM control of automatic-cleaning operations. For more information, see your LSM documentation.

- You can select a normal load or fast load option via the 97xx control panel. Not all tape management systems, however, support the fast load option.

Setting Scratch Preferences

On scratch mount requests you can explicitly specify the media type you want to use, or you can have ACSLS select a media type.

In order for ACSLS to select a media type, you must pre-define a prioritized list of compatible media types for each drive type. This list is referred to as the "scratch preferences."

For example, on TimberLine (9490) drives, you may want to use 3490E cartridges before 3490 or 3480 ones. To do this, you would define 3490E cartridges as the highest priority for 9490 drives, followed by 3490 and 3480 cartridges. Note that:

• There is one set of preferences for the entire server; preferences are not defined by client.

• If a compatible media type is not listed for a drive, the media will not be selected.

The following sections describe the user- and system-defined files that ACSLS uses in determining scratch preferences.

User-Defined Mixed-Media Files

Table 8 describes the user-defined mixed-media files that are located in `$ACS_HOME/data/external/mixed_media/`:

Table 8. User-Defined Mixed-Media Files

File	Description
`scratch_preferences.dat`	User-defined preferences file. Primary source of preference definitions.
`scratch_preferences.SAMPLE`	Sample preferences file. Can be copied to create the scratch_preferences.dat file.

System-Defined Mixed-Media Files

Table 9 describes the system-defined mixed-media files that are located in `$ACS_HOME/data/internal/mixed_media/`:

Hint: You cannot modify the system-defined mixed-media files.

Table 9. System-Defined Mixed-Media Files

File	Description
`media_compatibility.dat`	System-defined preferences file. Used only if the user-defined file does not exist or is missing a drive type.
`drive_types.dat`	System-defined list of supported drive types.
`media_types.dat`	System-defined list of supported media types.

How ACSLS Uses the Mixed-Media Files

Table 10 describes how ACSLS uses the mixed-media files to select a media type for a scratch mount request.

Table 10. How ACSLS Uses Mixed-Media Files

If the `scratch_preferences.dat` file ...	ACSLS does this...
Does not exist.	Uses the definitions in the system file, `media_compatibility.dat`.
Lists more than one media type for a drive.	Selects the media types in the order listed.
Lists no media type for a particular drive.	Uses the data in the system file, `media_compatibility.dat`.
Does not list a particular drive type.	Uses the data in the system file, `media_compatibility.dat`.

Defining a Scratch Preferences File

Use this procedure to define a `scratch_preferences.dat` file, which contains an ordered list of scratch cartridge types to select for given drive types. ACSLS uses this file for `mount *` command where a media type is not explicitly specified. Figure 5 shows the contents of the `scratch_preferences.SAMPLE` file.

Figure 5. scratch_preferences.SAMPLE

```
Drive Type NameMedia Type Preference Name
4480  3480
4490  3490E
9490  3490E
SD3   DD3A
SD3   DD3B
SD3   DD3C
```

To define a scratch preferences file, do the following:

1. Log in as `acsss`.

2. Change to the external mixed media directory:

   `cd $ACS_HOME/data/external/mixed_media`

3. Copy the sample scratch preferences file to create the user-defined file:

   `cp scratch_preferences.SAMPLE scratch_preferences.dat`

4. Using a text editor such as `vi`, modify the list of preferences in the scratch_preferences.dat file:

   • Follow the instructions in the comments at the top of the file.

   • If you want to use more than one media type for a drive type, enter each media type on a separate line; the order of preference is from top to bottom.

5. Save the file.

6. From a cmd_proc, restart ACSLS:

   `start`

See the command, .

■ Managing Dual TCP/IP Connections

Dual TCP/IP is an option that can be purchased for the SL8500 and SL3000. It provides two TCP/IP connections to the libraries. However, you can continue to use the libraries with only one of the two connections operational.

The purpose of dual TCP/IP is to automatically recognize and avoid a failing communication path. Since this is automated, there is no need for you to manually switch from an inoperative connection. For more information, see "ACSLS Dual TCP/IP Support" on page 344.

■ Managing a Dual-LMU Configuration

ACSLS supports *dual-LMU configurations*, which consist of:

• A *master* LMU that manages an ACS

• A `standby` LMU that automatically switches to master role to manage the ACS if the master LMU fails

Both LMUs are connected to a LAN that is connected to the LSMs. The first LMU powered on is initially the master, while the second LMU powered on is initially the standby. The LMUs periodically check each other's status so the standby can take over the master role if the master fails.

Note: ACSLS supports dual-LMU configurations for only the 9330 and L5530 LMUs with host/LMU microcode compatibility Level 12 (or above) loaded. The same microcode level must be loaded in both LMUs. ACSLS communicates with these LMUs by a serial connection, or through TCP/IP. There can be only one Ethernet connection for each TCP/IP connected LMU.

Limitation: ACSLS **does not** automatically initiate an LMU switch. Even if communication between ACSLS and the master LMU is lost, ACSLS does not initiate an LMU switch. ACSLS continues trying to communicate with the existing master LMU.

ACSLS dual-LMU support includes:

• ACSLS connects to both master and standby LMUs. ACSLS constantly monitors its connection to both LMUs. When communication is lost, ACSLS reports this condition.

• If the master LMU fails, the standby LMU automatically takes over as the new master LMU. When this happens, ACSLS recognizes the automatic switch and sends requests to the new master LMU. ACSLS also recovers transactions in progress (except for audits).

• You can issue the `switch lmu` command to manually switch ACS management from the master LMU to the standby LMU. ACSLS sends a "Force Switchover to Master" transmission to the standby LMU. The

standby takes over as the new master LMU. After a manual switchover, ACSLS recovers outstanding transactions (except for audits).

Hint: To further enhance the redundancy of a serial dual-LMU configuration, Sun StorageTek recommends that you use dual serial cables to connect the ACSLS server to *each* LMU. Figure 6 on page 70 shows an example of an ACSLS Sun server in a dual-LMU configuration. In Figure 6 on page 70, a port on the Sun connects to a serial patch panel, which has dual cabling to each LMU.

Figure 6. ACSLS Server in Dual-LMU Configuration
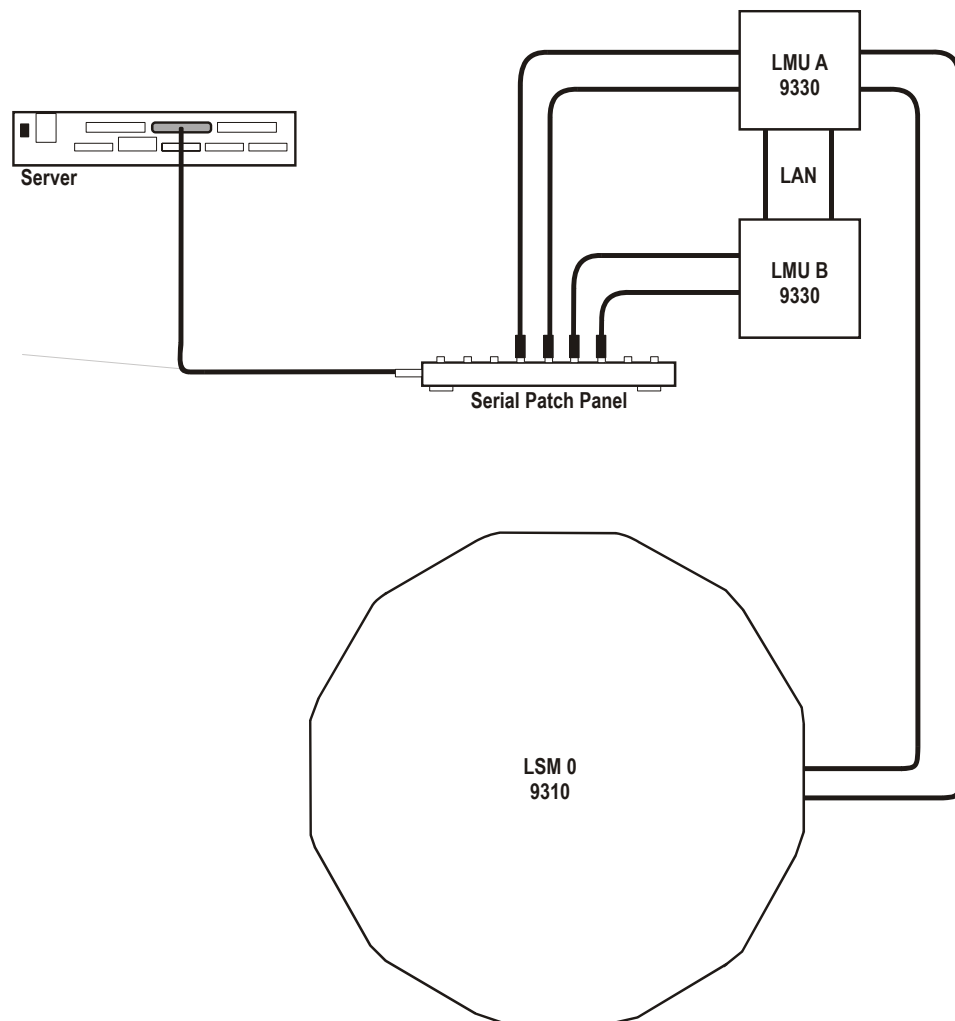
ACSLS dual-LMU support also enhances single-LMU configurations because the LMU can IPL without bringing down the ACSLS server.

The `query lmu` command displays LMU and port status for both single-LMU and dual-LMU ACS configurations; for more information see "query lmu" on page 238.

You can also use the `switch lmu` command to manually switch ACS management from the ACS's master LMU to the standby LMU; for more information see "switch lmu" on page 272. You can use manual switchover for hardware maintenance, such as replacing a LAN cable connecting dual LMUs. For example, assume that LMU A is in the master role and LMU B is in the standby role. If you need to replace a LAN cable, you can:

1. Switch to LMU B.

2. Vary the port(s) to LMU A offline.

3. Replace the LAN cable.

4. Vary the port(s) to LMU A back online.

5. Switch back to LMU A.

After an LMU switchover (automatic or manual), ACSLS recovers all outstanding (active and pending) requests. The time to complete each outstanding request during a switchover is increased by the time required to switch between LMUs and recover any preceding outstanding requests. Request recovery, therefore, can take three to five minutes.

■ Managing a Dual-LAN Client Configuration

ACSLS 5.2 and above supports *dual-LAN client configurations*, which consist of a primary LAN and a secondary (backup) LAN. If the primary LAN fails, the client switches to the secondary LAN. ACSLS removes all outstanding messages to that client and begins communicating via the secondary LAN. Because ACSLS removes all outstanding messages before switching over to the secondary LAN, a client will not receive a success message even though a request completed successfully.

For example, a client requests ACSLS to eject ten cartridges. ACSLS starts ejecting the cartridges, then communications fail between ACSLS and the primary client LAN. ACSLS removes all outstanding messages to that client and begins communicating via the secondary LAN. ACSLS successfully completes ejecting all ten cartridges, but sends no success message to the client. The client must verify that the request completed successfully. In this example, if the client issues a query volume request against the IDs of the ejected volumes, ACSLS returns a `volume not found` error message, which confirms that ACSLS ejected the cartridges.

ACSLS ensures that any transient requests complete successfully if a LAN communications switchover occurs. After switchover, however, the client must cancel any persistent outstanding requests (such as CAP operations) originally submitted on the primary LAN and resubmit these requests via communications from the secondary LAN. Resources (such as, CAPs, locks, drives, and so forth) allocated via primary LAN communications will remain allocated after switchover to the secondary LAN.

The following sections tell how to configure ACSLS for dual-LAN client operations by:

- Register the IP addresses of primary and secondary LANs by creating a `csc_ip_switch.dat` file

- Install a second ethernet port for a multi-homed ACSLS server

- Set the TCP/IP connection timeout interval to decrease the system switchover time to the backup LAN.

Registering the IP Addresses of Primary and Secondary LANs

To register the IP addresses of the primary and secondary LANs with ACSLS, create the file `csc_ip_switch.dat` in the `$ACSSS_HOME/data/internal/client_config/` directory. Figure 7 shows an example of a csc_ip_switch.dat file.

Figure 7. Example of csc_ip_switch.dat File

```
#The following entry is System Zed's primary and secondary LAN IP
addresses.
129.80.30.40129.80.30.50
```

As Figure 7 shows, use a # sign to precede comments. The entries consist of a client system's primary LAN IP address in the left column followed by one or more blanks, then a client system's secondary LAN IP address in the right column. In this example, System Zed's primary LAN IP address is 129.80.30.40 and its secondary LAN IP address is 129.80.30.50.

After you create or update the `csc_ip_switch.dat` file, if ACSLS is running, you must stop and restart ACSLS. If ACSLS is not running, start ACSLS. For more information, see "Stopping ACSLS" on page 3 and "Starting ACSLS" on page 2.

If ACSLS successfully reads the `csc_ip_switch.dat` file on restart, ACSLS logs the success message `2010 I DUAL PATH OPTION ACTIVATED` in the he ACSLS event log. Otherwise, dual-LAN support is not activated.

Installing a Second Ethernet Port for a Multi-Homed ACSLS Server

This section describes the procedure for installing a second ethernet port for a multi-homed ACSLS server. The second ethernet port provides the attachment to the second control path adapter, which controls the backup

LAN. This procedure requires the Sun SBUS Buffered Ethernet card. You can order this Ethernet card from Sun as part number X1053A or from Sun StorageTek as part number 309479501.

The installation procedure includes:

- Installing the hardware and rebuilding the kernel

- Defining a host name for the new ethernet port

- Creating the `/etc/notrouter` file

To install a second ethernet port, do the following:

1. Install the Sun SBus Buffered Ethernet card according to the manufacturer's instructions.

2. Reconfigure the system kernel for the new device:

    a. Power the system on. When it begins to boot, press [[STOP]]-[[A]] to enter the PROM monitor.

    b. At the ok prompt, boot the server:

       boot -r

3. When the boot has completed, login as root.

4. Create a hostname for the second ethernet port.

   `echo` *2nd_host_name* `> /etc/hostname/hme1`

   Where *2nd_host_name* is the hostname for the second ethernet port.

5. Enter the following:

   `touch /etc/notrouter`

   This creates the `/etc/notrouter` file.

6. Reboot the server:

   `reboot`

This completes this procedure.

■ Setting the TCP/IP Connection Timeout Interval

The UNIX system variable `tcp_ip_abort_cinterval` sets the TCP/IP connection timeout interval between the client and the ACSLS server. Changing the default value (180 seconds) of this variable may decrease the switchover time to the backup LAN. Note, however, that actual switchover time depends on the configuration and type of failure. For example, if an ACSLS server port fails and multiple clients are communicating via this port, ACSLS serially recovers communications with each client. Multi-client recovery for a

failing port therefore takes longer than recovery with a single client communicating with a failing port.

To decrease the switchover time to the backup LAN, do the following:

1. Log in as `root` on the ACSLS server.

2. From the prompt, enter the following:

   /usr/sbin/ndd -set /dev/tpc tcp_ip_abort_cinterval 15000

   This command changes the TCP/IP connection timeout interval to 15 seconds (the default is 180 seconds).

Hint: For Solaris only: To make this command persistent (across server reboots), add the command to the `/etc/rc2.d/S69inet` file in the "Set configurable parameters" section.

■ Registering for Event Notification

Event notification allows tracking of events occurring in tape libraries for ACSAPI clients. This feature is provided by the CSC Toolkit. Specifically, with event notification, the ACSAPI client can do the following:

- Register for library resource events and/or volume events

- Unregister for these events

- Check registration status for library events and be notified when they occur

Client registration requests and event notification messages are delivered to the client until the registration is dropped. Event notification periodically checks the registration status of the client to verify that the client is alive. This avoids sending responses to clients that are no longer active and avoids the unnecessary use of network resources.

The following types of events can be tracked:

- Addition and deletion of volumes and when maximum usage is exceeded for cleaning cartridges in the ACSLS database.

- Changes in library component status such as an LSM or drive changing from online to offline, diagnostic, or recovery; or a CAP that is opened or closed.

- Hardware failures such as an inoperable robotic hand.

Cartridge Management

7

Cartridge management consists of the following:

■ Using the Manual Volume Delete Utility

The Manual Volume Delete utility, `del_vol`, allows you to access a volume that is in an offline, unavailable LSM. If you manually remove the cartridge from the LSM and try to re-enter it into another LSM, ACSLS will issue a `duplicate volume` message and will not enter the cartridge. Using the `del_vol` utility, you can first delete the volume from the database, then manually remove it from the offline LSM and successfully re-enter it into an online LSM.

The Manual Volume Delete (del_vol) utility now retains volumes as absent, with the option to delete the volume. The volume can be deleted without waiting for the expiration of an absent or ejected status.

Notes:

- To remove a cartridge from an online LSM, issue an eject command for the cartridge.

- ACSLS and the database must be up and running to use this utility. Do not run `del_vol` while the system is in recovery; unpredictable results may occur.

For more information about this utility, see "del_vol" on page 143.

Note:

To delete a cartridge using the del_vol utility:

1. Log in as `acsss`.

2. Delete the cartridge:

   `del_vol` *vol_id*

For more information see about switch options you can use with `del_vol`, see "del_vol" on page 143.

■ Using Absent and Ejected Cartridge Support

Absent cartridge support in ACSLS marks cartridges that cannot be found in the library as absent, instead of deleting them. If these cartridges are later found in the library, ACSLS changes them to active status instead of re-adding them to the database. Reactivation preserves settings, such as pool, volume access control ownership, and locks.

Similarly, ejected cartridge support retains cartridge information when cartridges are ejected. The cartridges are reactivated when they are re-entered.

Absent and ejected volume support is enabled when the ABSENT_VOLUME_RETENTION_PERIOD is set to a non-zero number of days. The default value is 5 days.

Additional aspects of absent and ejected cartridge support include:

- The Manual Volume Delete (`del_vol`) utility will retain volumes as absent unless the `-d` option is specified. If this option is specified, the volume is deleted without waiting for the expiration of an absent or ejected status.

- ACSLS improves volume recovery by searching for volumes that are not found in their expected locations in the library. ACSLS searches all recorded locations instead of automatically deleting the volume.

- Clients can specify, through the ENABLE_STATUS_VOLUME_ABSENT and ENABLE_STATUS_VOLUME_MISSING configuration settings,

whether they want absent, ejected, and missing statuses to be reported through the ACSAPI.

• The `volrpt` utility with the `-i` option will report volume records with a status of absent or ejected. By default, volrpt does not report absent or ejected volumes.

■ Absent, Ejected and Missing Cartridges

ACSLS reports three cartridge (volume) statuses:

• missing

  The cartridge cannot be located in the library, and at least one recorded location for the cartridge cannot be searched because the LSM is offline or a drive is not communicating. The information about the cartridge has been retained.

• absent

  The cartridge cannot be located in the library. All recorded locations for the cartridge have been searched, and the cartridge is not in any of them. The information about the cartridge is retained. If the cartridge is found or re-entered into the library (before the retention period expires), it is reactivated.

• ejected

  The cartridge was ejected. The information about the cartridge is retained, and if the cartridge is found or re-entered (before the retention period expires), it is reactivated.

Cartridge (Volume) Status Reporting

ACSLS reports cartridges (volumes) with the status "missing", "absent," or "ejected," differently in response to ACSLS commands than it does in response to ACSAPI requests.

The information displayed in response to ACSLS commands identifies a cartridge as "missing", "absent", or "ejected."

However, the cartridge status information ACSLS displays in response to ACSAPI requests is governed by the following ACSLS dynamic variables:

1. missing

   a. If the ACSLS dynamic variable *ENABLE_STATUS_VOLUME_MISSING* is TRUE, ACSLS reports: *STATUS_VOLUME_MISSING.*

   b. If the ACSLS dynamic variable *ENABLE_STATUS_VOLUME_MISSING* is FALSE, ACSLS reports: *STATUS_VOLUME_IN_TRANSIT.*

2. absent

   a. If the ACSLS dynamic variable *ENABLE_STATUS_VOLUME_ABSENT* is TRUE, ACSLS reports: *STATUS_VOLUME_ABSENT*

   b. If the ACSLS dynamic variable *ENABLE_STATUS_VOLUME_ABSENT* is FALSE, ACSLS treats the volume as if it had been deleted from the ACSLS database and reports: *STATUS_VOLUME_NOT_IN_LIBRARY.*

3. ejected

   a. If the ACSLS dynamic variable *ENABLE_STATUS_VOLUME_EJECTED* is TRUE, ACSLS reports: *STATUS_VOLUME_EJECTED*

   b. If the ACSLS dynamic variable *ENABLE_STATUS_VOLUME_EJECTED* is FALSE, ACSLS treats the volume as if it had been deleted from the ACSLS database and reports: *STATUS_VOLUME_NOT_IN_LIBRARY.*

ABSENT_VOLUME_RETENTION_PERIOD Dynamic Variable

The *ABSENT_VOLUME_RETENTION_PERIOD* dynamic variable controls how long absent and ejected volumes are retained in the ACSLS database and specifies the number of days these volumes are retained. There are two special values:

- The value 0 (zero) days specifies that volumes are deleted and will not be marked absent or ejected. (This is the behavior of ACSLS releases previous to ACSLS 6.1.)

- The value 999 days specifies that absent and ejected volumes are retained forever in the database.

■ Cartridge Recovery

Cartridge Recovery (acscr) is an ACSLS internal process that is called to resolve discrepancies whenever the actual content of a storage cell or tape drive does not match the information saved in the ACSLS database. It does this by:

- Having the library examine the volume's home cell and, possibly, the drive, and then updating the ACSLS database with the results.

- With the SL3000 library (and soon the SL8500), ACSLS will recover a cartridge by asking the library where a cartridge is located, and then using the library's reply to update the ACSLS database.

If Cartridge Recovery finds a discrepancy (e.g., a cartridge that is recorded at another location), it creates another recovery request and adds it to its request queue. (This is called a "cascade".)

Other processes pass recovery requests to Cartridge Recovery when they encounter a discrepancy between the ACSLS database and the actual contents of the library. Because of this, Cartridge Recovery is the central location where cartridges are marked as missing, changed to absent, and reactivated. Thus, what appears to be the behavior of many other ACSLS commands and utilities is actually done by Cartridge Recovery when it updates the database to match the information reported by the library.

When other processes pass recovery requests to Cartridge Recovery, they can:

1. continue and let Cartridge Recovery continue asynchronously (Cartridge Recovery proceeds independently), or

2. if they need a specific cartridge that ACSLS can't locate, wait for Cartridge Recovery to finish processing this recovery request and report what it found.

■ Missing Cartridges

A cartridge is marked missing when:

- Cartridge Recovery cannot find a cartridge in the library, and

- it cannot examine all recorded locations for a cartridge (home cell and drive, if the cartridge has a recorded drive location).

For example, when Cartridge Recovery cannot examine the home cell in an offline LSM or an offline drive, and when it doesn't find the cartridge in other locations, it marks the cartridge missing.

Cartridge Recovery preserves the cartridge's home location unless it examines the cartridge's home cell and finds another cartridge there. In this situation it marks the cartridge "homeless," with a minus one (-1) in the home_lsm field.

When Cartridge Recovery finds a cartridge that was missing, it changes that cartridge's status to "home" or "in drive" in the database, depending where it found the missing cartridge.

1. If the cartridge is found in a cell other than its recorded home cell, Cartridge Recovery checks the cartridge's home cell to see whether it has found a duplicate cartridge.

2. If the cartridge is not in its recorded home cell, Cartridge Recovery records the cell in which it was found as its new home cell.

3. If the new cartridge is a duplicate, Cartridge Recovery reports this on the Event Log. The duplicate cartridge is **not** ejected.

4. If Cartridge Recovery finds a "homeless" cartridge in a drive, it does not assign a new home cell. When the cartridge is dismounted, the dismount process assigns a new home cell.

■ Absent and Ejected Cartridges

Cartridges Not Found

When Cartridge Recovery can examine all recorded locations and it cannot find a cartridge:

1. If the *ABSENT_CARTRIDGE_RETENTION_PERIOD* is 0, Cartridge Recovery

   • deletes the cartridge record from the database.

   • marks the cell record in the database for the cell that was the cartridge's home cell as "empty."

2. If the *ABSENT_CARTRIDGE_RETENTION_PERIOD* is greater than 0, Cartridge Recovery

   • changes the status of the cartridge record in the database to "absent" if the cartridge has not already been marked absent or ejected.

   • records the cartridge as "homeless" (with a minus one (-1) in the home_lsm field).

   • marks the cell record in the database of the cartridge's former home cell as "empty."

Cartridges Found

If Cartridge Recovery finds an ejected or absent cartridge, it reactivates the cartridge.

If the ejected or absent cartridge is found in a storage cell, this becomes its new home cell, and Cartridge Recovery changes the cartridge's status to "home" in the database.

If the cartridge is found in a drive, ACSLS assigns a new home cell when the cartridge is dismounted.

■ Populating the LSM

An essential requirement for the library and ACSLS to function properly is the availability of a few free cells in each LSM to accommodate dismount, pass through, and eject operations. You should reserve at least one free cell for each tape drive installed in each LSM.

To determine the free cell count of an LSM, issue the command:

query lsm *lsm_id*

Note: In the SL8500, each rail is defined as an LSM.

■ Using the CAP

CAP Types

Each type of CAP has a standard capacity and method for loading it with cartridges. An LSM may have more than one type of CAP. Table 11 shows the supported CAP types, identifiers and capacities, and loading methods.

Table 11. CAP Types

CAP Type	Identifier & Capacity	Loading Method
Standard	CAP00; holds 21 cartridges.	Cartridges are loaded directly into the CAP cells.
Enhanced (4410 and 9310)	CAP00 and CAP01; each holds 40 cartridges.	Cartridges are placed into removable magazines which are loaded into the CAP.
9360	CAP00 holds 20 cartridges; optional CAP01 holds 30 cartridges.	Cartridges are placed into removable magazines which are loaded into the CAP.
Priority (PCAP)	CAP02; holds one cartridge.	Cartridges are entered one at a time, directly into the CAP.
9710 or 9740 CAP	CAP00, holds 14 cartridges or magazine, which holds 10 cartridges.	Cartridges are loaded directly into the CAP cells or placed into the removable magazine which is loaded into the CAP.

Table 11. CAP Types

CAP Type	Identifier & Capacity	Loading Method
9714, 9730, or 9738 CAP	CAP00, holds one cartridge	Cartridges are loaded directly into the single-cell CAP.
L180	CAP00, holds 10 cartridges.	5 cartridges are placed into each of 2 removable magazines which are loaded into the CAP.
L700	CAP00 and optional CAP01; each holds 20 cartridges.	5 cartridges are placed into each of 4 removable magazines which are loaded into the CAP.
SL500	CAP00; holds between 5 and 25 cartridges	5 cartridges are placed into removable magazines which are loaded into the CAP. One magazine in base module; 2 magazines in expansion modules that contains a CAP.
SL8500	CAP00 and optional CAP01; each holds 39 cartridges.	13 cartridges are placed into each of 3 removable magazines which are loaded into the CAP.
SL3000	CAP 06 and optionally CAP 01-05, CAP 07-10; each holds 26 cartridges.	13 cartridges are placed into each of 2 removable magazines which are loaded into the CAP.

CAP States

A CAP's state determines whether it is available for entering and ejecting cartridges. Table 12 describes the valid CAP states. See "Displaying CAP Information" on page 87 for procedures for determining the CAP state. See the command, "vary" on page 277 for information about changing device states.

Note: Refer to "SL8500 CAP Behavior" on page 322 for specifics regarding the LS8500 library. Refer to "SL500 CAP Behavior" on page 380 for specifics regarding the SL500 library.

Table 12. CAP States

State	Description	How requests are handled
diagnostic	The CAP is available for diagnostic activity without interference from client applications.	• Requests from client applications are rejected. • Requests from the cmd_proc are processed.
offline	The CAP is logically disabled.	• All requests are rejected.
offline-pending	Transition state. Occurs when the CAP is taken from online to offline.	• All new requests are rejected. • Current and pending requests are processed to completion.
online	Normal operating state.	• All requests are accepted and processed.
recovery	Transition state. Occurs when the CAP is taken from offline to online.	• New requests are rejected.

CAP Modes

The CAP mode controls how a CAP will be used for cartridge enters and ejects. Table 13 describes the valid CAP modes. "Displaying CAP Information" on page 87 for procedures for determining the CAP mode. See the command, "set cap mode" on page 260 for information about changing the CAP mode.

Hint: You cannot change a CAP's mode while the CAP is in use. That is, if the door is open during either manual or automatic enter operations, you cannot change its mode until you complete the enter operation.

316120101

Table 13. CAP Modes

Mode	Description	Effects on enter/eject

automatic	The CAP is unlocked when not in use. This is the initial mode for all priority CAPs.
	
	Note: You cannot set the CAP mode to automatic in a partitioned library. The exception to this is, a dedicated CAP in the SL3000 can be set to automatic mode.
		• You can enter cartridges without explicitly issuing an enter command. The enter is initiated when you open the CAP door, place the cartridge(s) inside, and close the CAP.
		
		Note: You cannot cancel an automatic enter operation that is in progress using the cancel command. To terminate an automatic enter in progress:
		
		If the CAP door is open, remove all the cartridges and close the door.
		
		If the CAP door is closed and the cartridges are being moved into the library, you must allow the remaining cartridges to be entered into the library. The enter then terminates.
		
		• To eject cartridges you must explicitly issue an <u>e</u>ject command. You can either specify the *cap_id* on the command or allow ACSLS to automatically select a CAP, based on previously defined CAP priorities.
		
		ACSLS shows the CAP in automatic mode, but it cannot be opened and used for automatic enters. You must synchronize ACSLS and the SL8500, and return the CAP to automatic enter. To do this:
		
		`set cap manual cap_id`
		`set cap automatic cap_id`

	Note: When the SL8500 access door is opened and closed, the SL8500 leaves the CAP locked. When the CAP is locked, it cannot be used for automatic mode enters.	

Table 13. CAP Modes

Mode	Description	Effects on enter/eject
manual	The CAP is locked when not in use. This is the initial mode for all multi-cartridge CAPs.	• You can enter or eject cartridges only after explicitly issuing a command. You either specify the *cap_id* on the command, or allow ACSLS to automatically select a CAP, based on previously defined CAP priorities.

Note: Some client applications require CAPs to be in manual mode. See the documentation for your tape management system.

CAP Priorities

CAP priorities specify how ACSLS automatically selects CAPs when the CAP request specifies an asterisk (*) for the CAP ID, The following table describes the CAP priorities and their effect. See for procedures for determining the CAP priority. See for information about changing the CAP priority.

Table 14. Cap Priorities

Priority	Effect
16 (highest)	first used
15 (next highest)	next used
-	
-	
-	
1 (lowest)	last used
0	never automatically selected (initial priority for all CAPs)

CAP priorities and automatic CAP selection apply to the following commands:

• audit

• eject

• enter

• venter

When you enter any of these commands with an asterisk (*) as the *cap_id*, ACSLS automatically selects an available CAP with highest non-zero priority for each ACS specified in the request.

Displaying CAP Information

Following are some guidelines for using the `query cap` command to display current CAP information.

To display information for selected CAPS, enter:

query cap *cap_id cap_id ...*

To display information for all CAPS in the library, enter:

query cap all

Figure 8 shows a typical example of the results of a query cap command.

Figure 8. Example of Query CAP command

```
ACSSA> query cap 0,0,0,0,0,1,0,1,0

2004-06-30 13:15:46              CAP Status
Identifier priority Size  State    Mode       Status

000        1        21    online   automatic  enter
001        0        4     online   manual     eject
010        7        21    offline  automatic  available
```

■ Recovering CAPs

General CAP Recovery Procedures

Recover CAP by Varying Offline Then Online

1. Vary the CAP offline.

   vary cap *cap_id* offline

   Note: If you have to force the CAP offline:

   vary cap *cap_id* offline force

   Only the current robotic request completes, then the CAP goes offline immediately. Pending requests are discarded and new requests are rejected.

2. Vary the CAP back online.

   vary cap *cap_id* online

CAP in Auto Mode and ACSLS not Synchronized with Library

To synchronize the state of the CAP between both ACSLS and the library when the CAP is in automatic mode:

1. Set the CAP mode to manual to terminate automatic enter mode.

    ```
    set cap mode manual cap_id
    ```

2. Set the CAP back to automatic mode.

    ```
    set cap mode automatic cap_id
    ```

CAPs Must be Empty Before Enters and Ejects Terminate

ACSLS requires that a CAP be empty before it is available for enters and ejects. During initialization, or when an ACS or LSM comes online, ACSLS requests that you remove any cartridges found in a CAP. All cartridges must be removed from the CAP before ejects terminate.

During an enter, if cartridges with missing or invalid external labels are present, they are not moved into the library. The CAP is unlocked, and you must remove the invalid cartridges before the enter can proceed.

Note: With a 9310 and other libraries, the robot stops scanning when it encounters an empty CAP cell. A CAP with only the first cell empty appears to be completely empty.

Recovery Procedures to Unlock a CAP in an L1400, L700, L700e, or L180 Library

If a CAP that is used for an enter or eject in an L1400, L700, L700e, or L180 library does not unlock, you can IPL the library to recover the CAP. Follow the appropriate procedure below to recover the CAP.

CAP used for a manual enter will not unlock

To recover a CAP that will not unlock for a manual enter:

1. Cancel the enter.

    Use either [Control + C] at the *cmd_proc* that is waiting for the enter, or cancel the enter request ID.

2. Re-IPL the library by pressing the RESET button on the operator panel.

3. Start another enter after the library finishes initialization.

CAP used for an automatic enter will not unlock

To recover a CAP that will not unlock for an automatic enter:

1. Set the CAP mode back to manual to terminate automatic enter mode.

   ```
   set cap mode manual cap_id
   ```

2. Re-IPL the library by pressing the RESET button on the operator panel.

3. Set the CAP back to automatic mode after the library finishes initialization.

   ```
   set cap mode automatic cap_id
   ```

Cannot empty CAP used for an eject because it will not unlock

To recover a CAP that will not unlock for an eject (after the CAP is full or all volumes have been ejected):

1. Open the access door to the library, remove all cartridges from the CAP, and close the access door.

2. Re-IPL the library by pressing the RESET button on the operator panel.

   Note: A Re-IPL of the library causes ACSLS to terminate the eject with a "library failure".

3. Optionally, audit the library.

   After the library finishes initialization, running an audit is a good idea, but it is not required.

4. Start another eject if all the cartridges were not ejected.

Recovering a CAP After Opening the SL8500 Access Door

When the SL8500 access door is opened and closed, the SL8500 re-initializes. After this re-initialization, the CAP is locked (regardless of whether it was unlocked before).

Note: When a 9310 re-initializes, if the CAP was unlocked before, it remains unlocked afterwards.

After the SL8500 re-initializes, if the CAP is locked and you need to recover it, follow the appropriate procedure below to recover the CAP.

CAP used for automatic enter will not unlock

To recover a CAP that will not unlock for an automatic enter, you must synchronize the state of the CAP between both ACSLS and the library: To do this:

1. Set the CAP mode to manual to terminate automatic enter mode.

   ```
   set cap mode manual cap_id
   ```

2. Set the CAP back to automatic mode.

   ```
   set cap mode automatic cap_id
   ```

CAP used for manual enter will not unlock

To recover a CAP that will not unlock for an manual enter, you must synchronize the state of the CAP between both ACSLS and the library. To do this:

1. Vary the CAP offline.

   ```
   vary cap cap_id offline
   ```

2. Vary the CAP back online.

   ```
   vary cap cap_id online
   ```

3. Restart your manual enter.

   ```
   enter cap_id
   ```

CAP used for ejects will not unlock

To recover a CAP where you were doing an eject, you must remove any cartridges left in the locked CAP and synchronize the state of the CAP between both ACSLS and the library. To do this:

1. Remove any cartridges in the CAP. To do this:

   a. Vary the CAP offine.:

   ```
   vary cap cap_id offline
   ```

   b. Vary the CAP back online.

   ```
   vary cap cap_id online
   ```

2. Select one of the following:

   If the CAP is in auto mode:

   a. Set the CAP mode to manual to terminate the automatic enter mode.

   ```
   set cap mode manual cap_id
   ```

   b. Set the CAP to automatic mode. This unlocks the CAP.

   ```
   set cap mode automatic cap_id
   ```

   c. Remove any cartridges left in the CAP.

   If the CAP is not in auto mode:

   a. Start a manual enter.

   ```
   enter cap_id
   ```

   b. Remove any cartridges left in the CAP.

   c. Cancel the enter.

Use either [Control + C] at the *cmd_proc* that is waiting for the enter, or cancel the enter request ID.

3. Restart your eject.

    ```
    enter cap_id  vol_id|volrange…
    ```

■ Entering Cartridges

You can choose to enter cartridges manually or automatically.

*   To enter cartridges manually, you need to issue the `enter` command.This unlocks the CAP so cartridges can be entered.

*   An automatic enter is initiated by opening a CAP that is in automatic mode. When a CAP is in automatic mode, you need not issue an enter command.

The following steps describes the enter process:

1. Once you start the enter, the CAP is unlocked and reserved. It cannot be used by another host.

2. After you open the CAP, place the cartridges into the CAP, and close the CAP, the CAP is locked.

    The ACSLS library robot inspects/audits the cartridges in the CAP. All cartridges being entered must have valid external labels that do not duplicate other `vol_ids` already managed by this ACSLS server.

    Note: Virtual enter allows you to enter unlabeled cartridges into some libraries.

3. ACSLS allocates home cells in the library to valid cartridges and moves them to their assigned home cell locations.

    Duplicate cartridges and cartridges without external labels are left in the CAP and must be removed.

4. Upon completion, the CAP is unlocked so more cartridges can be entered.

    *   If the CAP is in automatic mode, the automatic enter is complete and the CAP is unreserved and available.

    *   If this is a manual enter, the CAP is still reserved for the manual enter. To terminate the manual enter, cancel it with either the cancel command or with <CTRL>+C at the `cmd_proc` where the enter was started.

For additional information on the enter command, refer to .

Note: If cartridge tracing is enabled, the event log records all cartridge enters.

Table 15. Cartridge Enter Commands

Task	Command
Entering Cartridges in Automatic Mode	`set cap mode automatic cap_id`
Entering Cartridges in Manual Mode	`enter cap_id`
Entering Cartridges with Virtual Labels (venter)	`venter cap_id vol_id vol_id`
	Do *not* open the LSM door and place cartridges with missing or unreadable labels in a storage cell because ACSLS cannot manage these cartridges. During an audit, ACSLS will eject cartridges with missing or unreadable labels that were ventered.

Terminating an Enter Request

Use these procedures to terminate or cancel a current or pending manual enter or virtual enter.

Note: You cannot cancel an automatic enter operation that is in progress using the cancel command. To terminate an automatic enter in progress:

-   If the CAP door is open, remove all the cartridges and close the door.

-   If the CAP door is closed and the cartridges are being moved into the library, you must allow the remaining cartridges to be entered into the library. The enter then terminates.

To cancel a manual enter

1. Display all current and pending library activity:

    `query request all`

2. Note the *request_id* of the enter/venter request you want to cancel.

3. From the `cmd_proc`, enter:

    `cancel request_id`

    where *request_id* is the identifier of the request you want to cancel.

4. Wait for the CAP to unlock, open the CAP, and remove all cartridges.

   The cmd_proc will display a message indicating the number of cartridges entered into the library before the cancel request was received. These cartridges will remain under ACSLS control.

See also:

- "query request" on page 253
- "cancel" on page 194

Ejecting Cartridges

To eject cartridges from the library you need to issue the eject command.

The following steps describes the eject process:

1. Once you start the eject, the CAP is locked. It cannot be used by another host.

2. The robot places the specified cartridge(s) in the designated CAP, then ACSLS makes the cell locations where the cartridges were stored available for other cartridges.

3. Open the CAP, remove all cartridges from the CAP, and close the CAP. ACSLS then inspects the CAP to ensure it is empty. The CAP is now available for another operation, such as enter or audit.

   Note: If you specify more than a CAP full of cartridges on the eject command, empty the CAP when it fills, close the CAP, and ACSLS continues the eject process until all cartridges are ejected.

For additional information on the eject command, refer to "eject" on page 207.

Note: If cartridge tracing is enabled, the event log records all cartridge ejections.

Managing Scratch Cartridges

Scratch cartridges either contain no data or data that can be overwritten. A user or application mounts a scratch cartridge to write new data on that cartridge.

You must ensure that the library has enough available scratch cartridges to satisfy scratch mount requests. For more information see:

- "Adding Scratch Cartridges to the Library" on page 94
- "Rebalancing Scratch Pools" on page 95

The following sections provide additional information about managing scratch cartridges and scratch pools:

Displaying Scratch Pool and Scratch Cartridge Information

To display scratch pool information, use the following ACSLS functions:

`query pool`

> Displays scratch pool attributes, see "query pool" on page 250.

`query scratch`

> Displays scratch cartridge information, see "query scratch" on page 254.

`query mount *`

> Displays the status of media-compatible cartridges for a specified scratch pool (and, optionally, for a specific cartridge media type within the pool), see "query mount *" on page 247.

customized cartridge report

> Can be created to report selected scratch cartridge information, see "Creating a Custom Volume Report" on page 121.

Adding Scratch Cartridges to the Library

Use this procedure to add scratch cartridges to the library.

To add scratch cartridges to the library:

1. If necessary, create a new scratch pool:

   For more information, see the "define pool" on page 200

2. Enter scratch cartridges into the library.

   For more information, see "Entering Cartridges" on page 91

3. Define the cartridges you entered in Step 2 as scratch cartridges and assign them to a scratch pool.

   For more information, see "set scratch" on page 267

Rebalancing Scratch Pools

Use this procedure to rebalance scratch pools by moving scratch cartridges from one pool to another.

To rebalance scratch pools:

1. To display the attributes of all scratch pools, enter:

   query pool all

   For more information see "query pool" on page 250

2. Use the query scratch command to display the IDs of scratch cartridges in pools you want to rebalance.

   For more information, see "query scratch" on page 254

3. Use the set scratch command to move scratch cartridges from one pool to another.

   For example, to move cartridges YUMA20 through YUMA80 (which currently reside in pool 5) to pool 10, enter:

   set scratch 10 YUMA20-YUMA80

   For more information, see "set scratch" on page 267

Deleting Scratch Pools

To manage scratch pools, you may want to delete any scratch pools that no longer contain scratch cartridges. You cannot delete the common pool (Pool 0). Note that you can delete only empty scratch pools; you cannot delete a scratch pool if it contains either data or scratch cartridges. You can, however, use "Deleting All Empty Pools" on page 96 to delete all empty pools (ACSLS will not delete any pools that contain scratch or data cartridges).

Emptying a Scratch Pool

Use this procedure to empty a scratch pool before deleting it.

To empty a scratch pool:

1. To move data cartridges out of the pool, enter:

   set scratch off 0 *vol_id volrange ...*

   Where the *vol_id* or *volranges* specify the data cartridges you want to move to the common pool (pool 0). For more information, see "set scratch" on page 267.

2. To move scratch cartridges out of the pool, do one of the following:

- Move the cartridges to another pool.

- See "Ejecting Cartridges" on page 93. If you eject scratch cartridges, however, ACSLS no longer manages these cartridges. If you later want to use these cartridges, you must reenter them and assign them to a scratch pool.

Deleting a Single Pool

To delete a single pool:

```
delete pool pool_id
```

Deleting All Empty Pools

The <u>delete pool all</u> command deletes only empty scratch pools, not pools that contain scratch or data cartridges.

To delete all empty pools:

```
delete pool all
```

Mounting Scratch Cartridges

Use the following procedures to mount scratch cartridges in single-media and mixed-media environments.

Single-Media Environments

To mount a cartridge from a specified pool:

```
mount * drive_id pool_id
```

If no cartridge is available from the specified pool and the pool has been set for "overflow," ACSLS will select a cartridge from the common pool (pool 0).

To mount a cartridge from the common pool:

mount * *drive_id*

Mixed-Media Environments

To mount a scratch cartridge with a specified media type from a specified pool:

mount * d*rive_id pool_id media media_type*

If no cartridge is available from the specified pool and the pool has been set for <u>overflow</u> ACSLS will select a cartridge with the specified media type from the common pool (pool 0).

To mount a scratch cartridge from a specified pool with a media type determined by scratch preferences:

```
mount * drive_id pool_id media *
```

If no cartridge is available from the specified pool and the pool has been set for overflow ACSLS will select a cartridge from the common pool (pool 0) according to the defined scratch preferences.

To mount a cartridge from the common pool with a specified media type:

```
mount * drive_id media media_type
```

To mount a cartridge from the common pool with a media type determined by scratch preferences:

```
mount * drive_id media *
```

See also:

- "mount *" on page 223
- "query commands" on page 230
- "Setting Scratch Preferences" on page 66

Unscratching Cartridges

Use this procedure to "unscratch" cartridges (return them to data cartridge status) that were scratched in error.

To unscratch cartridges:

1. Use the query pool and query scratch commands to display the cartridge and pool IDs of the cartridges to unscratch.

   For more information see "query pool" on page 250 and "query scratch" on page 254.

2. To unscratch the selected cartridges, enter:

   ```
   set scratch off 0 vol_id volrange ...
   ```

   Where the vol_id or volranges specify the cartridges you want to change from scratch mode and move them to the common pool (pool 0). For more information, see "set scratch" on page 267.

■ Cleaning Cartridges

ACSLS controls automatic cleaning for HLI-attached libraries (SL8500, L5500, 9300, 9740, and 4400 serial or TCP/IP attached libraries), but not for SCSI-attached libraries.

Cartridges must be cleaned periodically to prevent damage to them and the tape media. Transport control units track how much tape passes through each transport and send a message to ACSLS when a transport requires cleaning.

Automatic Cleaning

If auto-cleaning is enabled, ACSLS automatically mounts a cleaning cartridge on the transport when required. If all cleaning cartridges are expired (the *max_usage* value is exceeded), ACSLS honors the original mount request without cleaning the transport. For that mount and for each subsequent mount to the uncleaned drive, ACSLS posts message 376N to the event log. Add more cleaning cartridges of compatible media with the drive type as described in "Defining Cleaning Cartridges" on page 98.

If auto-cleaning is disabled, ACSLS logs a message in the event log and displays cleaning messages at the cmd_proc when the transport requires cleaning. You then must manually mount a cleaning cartridge.

Use the acsss_config configuration program to enable or disable auto-cleaning. In addition, with acsss_config you can specify how cleaning cartridges are ordered for selections and queries. For more information, see Chapter 4, "Setting Variables that Control ACSLS Behavior."

Note: You cannot enable auto-cleaning on SCSI-attached libraries.

For more information on cleaning cartridges, refer to:

- "Defining Cleaning Cartridges" on page 98
- "Ejecting Used Cleaning Cartridges" on page 99
- "Manually Cleaning a Transport" on page 100
- "Cleaning Cartridges on SCSI-Attached LSMs" on page 101
- "Correcting Cleaning Cartridge Problems" on page 101

Defining Cleaning Cartridges

You must use the set clean command to define a cleaning cartridge because ACSLS does not define cleaning cartridges by *vol_id*.

When you define cleaning cartridges, make sure to:

- Use cleaning cartridges whose media types are compatible with each transport type in your library. ACSLS will automatically select the correct type of cartridge for each cleaning operation.

- Define at least one cleaning cartridge for each transport in your library.

To define cleaning cartridges to ACSLS:

1. Make the CAP ready for entry.

   See "Entering Cartridges" on page 91 for more information.

2. Enter the cleaning cartridges.

   The `cmd_proc` displays messages with the cartridge IDs of the cartridges you enter.

3. To set the cleaning cartridge attribute:

   `set clean` *max_usage vol_id | volrange*

   Where:

   max_usage is the number of times a cleaning cartridge is used before ACSLS stops selecting the cartridge to clean cartridges.

   vol_id|volrange specifies the cleaning cartridge or range of cartridges that you entered in Step 2.

See also:

- "enter" on page 211
- "set clean" on page 264
- "query clean" on page 235
- "volrpt" on page 175

Ejecting Used Cleaning Cartridges

ACSLS logs a message to the event log when a cleaning cartridge has reached the maximum use specified on the set clean command. ACSLS leaves the cartridge in the library, but will no longer select it for cleaning. Eject the used cleaning cartridge and enter a replacement.

To eject used cleaning cartridges:

1. To eject the cleaning cartridges, enter:

   `eject` *cap_id vol_id|volrange*

   Where:

   cap_id specifies the CAP used to eject the cleaning cartridges.

   vol_id|volrange specifies the IDs of the cleaning cartridges to eject.

2. Remove the cleaning cartridges.

See also:

- "query clean" on page 235
- "eject" on page 207
- "volrpt" on page 175

Manually Cleaning a Transport

Use this procedure to clean a transport when auto-clean is disabled or is not working.

To manually clean a transport:

1. Determine which cleaning cartridge types are compatible with the drive to be cleaned.

   Refer to the *Product Information Guide* and look in the Transport and Media Compatibility table for a list of the cleaning cartridges for each drive type

   If you do not have access to this manual, you can download it from the Customer Resource Center (CRC).

2. Display the available cleaning cartridges:

   ```
   query clean all
   ```

   To display all compatible cleaning cartridges in the same ACS as the drive, use the display command.

   ```
   display volume * -home acs,*,*,*,* -media media_type
   ```

   To display the cartridge's maximum cleaning usage and current usage:

   ```
   display volume * -home acs,*,*,*,* -media media_type
   -field vol_id acs lsm media max_use access_count
   ```

3. Select a compatible cleaning cartridge from those listed and mount it on the transport:

   ```
   mount vol_id drive_id
   ```

4. After the transport is cleaned and the cleaning cartridge is unloaded, dismount the cleaning cartridge:

   ```
   dismount vol_id drive_id
   ```

See also:

- "query clean" on page 235
- "display volume" on page 304
- "mount" on page 220
- "dismount" on page 204

Cleaning Cartridges on SCSI-Attached LSMs

You cannot use the `acsss_config` configuration program to enable auto-cleaning for drives attached to SCSI-attached LSMs. Using ACSLS, you can only clean these drives by manually mounting a cleaning cartridge. The LSM hardware, however, lets you enable auto-cleaning via the LSM control panel. For more information, see your LSM documentation. For 9714, 9730, or 9738 LSMs, if you either enable or disable auto-cleaning via the control panel, for the change to take effect you must do the following:

*   Rerun `acsss_config` without changing any options but ensuring that you enter `Y` to the `Build/verify library configuration?` prompt.

Correcting Cleaning Cartridge Problems

The following procedures tell how to:

*   Change a cleaning cartridge's maximum use count. For example, if the manufacturer recommends that you use a cartridge only 10 times and you set *max_usage* to 20, reset *max_usage* to 10 if the cartridge is unused. If the cartridge was used five times, reset *max_usage* to 5.

*   Set a cartridge's cleaning cartridge attribute off. For example, if you incorrectly defined a data cartridge as a cleaning cartridge, set the cartridge's cleaning cartridge attribute off to redefine the cartridge as a data cartridge.

To change a cleaning cartridge's maximum use count:

```
set clean max_usage vol_id | volrange
```

Where:

>   *max_usage* is the new maximum use.
>
>   *vol_id* | *volrange* specifies the cleaning cartridge or range of cartridges.

To turn off a cartridge's cleaning cartridge attribute:

```
set clean off vol_id | volrange
```

Where *vol_id* | *volrange* specifies the cartridge or range of cartridges.

■ Manually Loading Cartridges Into Drives In a Disabled LSM

If your LSM fails and you take it off line, you can still manually load cartridges into the library drives if the data path is still operational.

To manually load cartridges into drives in a disabled LSM:

1. Open the LSM door.

2. Note the cartridge labels of any cartridges that are already in drives and remove these cartridges.

   You will need to replace these cartridges at the end of this procedure.

3. Load the drives with the cartridges you want to read from or write to.

   Repeat this step as often as necessary until the LSM is repaired, then continue with Step 4.

   CAUTION: **In this step, you can remove cartridges from library cells and load these cartridges in the drives. Note the cell locations of these cartridges and ensure that you return the cartridges to these locations in Step 4.**

4. After the LSM is repaired, remove all cartridges from the drives and replace them with the original cartridges you noted in Step 2.

5. Close the LSM door, vary the LSM back online, and resume normal operations.

Database Administration

<div style="text-align: right;">8</div>

The database contains all information about the library configuration and the location of all library cartridges.

ACSLS control files that are backed up and are recovered include the customer-configurable files located in `$ACS_home` under `data/external` and some files associated located in `data/internal/client_config`.

This chapter discusses: importing and exporting the database; verifying the imported database and library configuration; backing up the database; and restoring and recovering the database.

- Exporting and Importing the database includes:

    - Exporting the database to a disk file or local tape device

    - Importing the database from a disk file or local tape device

    - Importing ACSLS control configuration files

    - Merging any customized dynamic variables

    - Verifying the imported database and library configuration

    CAUTION: **AIX only.**

    If you are using a tape device and migrating between AIX and Solaris platforms, you need to change the block size of the local tape device to 0.

    If the block size is not changed when exporting to Solaris or importing from Solaris, you are likely to see errors generated. To avoid this you need to take the following actions:

- **Export the ACSLS data from AIX to Solaris**

  To do this:

  - Change the block size.

    a. As `root on AIX`, use the appropriate tape device for your environment.

       Example:
       `chdev -l rmt1 -a block_size=0.`

    b. Verify that the block_size was changed to 0 by running the command that is appropriate for your environment.

       Example:
       `mt -f /dev/rmt1 status`

  - Export the data from AIX using `db_export.sh -f <tape_device>`. See "Manual Database Exports" on page 310.

  - Import the data to Solaris using `db_import.sh -f <tape_device>`. See "Importing the Database" on page 108.

- **Import ACSLS data from Solaris to AIX**

  To do this:

  - Export the data from Solaris using Option 6 of `db_export.sh -f <tape_device>`. See "Manual Database Exports" on page 310.

  - Change the block size as described above.

  - Import the data to AIX using any option of `db_import.sh -f <tape_device>`.

- Backing up the database and ACSLS control files includes:

  - Automatic database backup

  - Performing manual backups to a local tape device or to disk

  - Backing up to a UNIX File

  - Creating a backup that can be restored to a different server

- Recovering and restoring the database and ACSLS control files includes:

  - Restoring the database to the most recent backup

  - Recovering from a specific file

  - Restoring ACSLS control files

- Restoring a backup created on a different server

- Restarting the database

■ Utilities Used

You will use the following utilities:

- the `bdb.access` utility for backups to a:

    - specified UNIX file

    - tape device

    - default file and location

- the rdb.acsss utility for

    - recovering the database from corruption

    - from changes that produce unintended results

    - from server failure

- the `db_export.sh` and `db_import.sh` utilities for migrating between versions of ACSLS. This includes going to a later release or going to a previous release.

When you install ACSLS, you also automatically install the database management software. The ACSLS database is initialized after ACSLS is installed when you:

- configure the library hardware using `acsss_config`

- import a previous exported database using `db_import.sh`

- recover a database backup created on a different server using `rbd.acsss.sh`

■ Exporting the Database

This section describes how to migrate the ACSLS database and its associated ACSLS control files from either a previous version of ACSLS, the same release level of ACSLS, or return to a prior release.

The `db_export.sh` utility creates an ASCII representation of the database on tape or a specified file to disk. It is also responsible for gathering ACSLS control files. This utility can be used in two different ways.

- If it is executed without any options, the exported files are copied to the local tape drive.

- Executed with "-f" option followed by a path and tape device

- If it executed with "-f" option and proceeded by a path and file name, the exported files reside in the name file, and it's associated ACSLS control file, designated by the `.misc` extension.

The files generated by `db_export.sh` are then used as input to the `db_import.sh` utility at the time of an upgrade or recovery.

Note: This is the preferred method to use to migrate from all previous versions of ACSLS to the most current version.

When executing the `db_export.sh` utility either with the `-f` option or without, you will be prompted to choose the version of ACSLS to which you are exporting. Selections are:

```
1: ACSLS 5.3.2 or 5.4

2: ACSLS 6.0 or 6.0.1

3: ACSLS 6.0.1 with L700e

4: ACSLS 6.0.1 with PUT0201

5: ACSLS 6.1, 7.0, 7.1/7.1.1 before PUT0701

6: ACSLS 7.1/7.1.1 with PUT0701 or ACSLS 7.2 (any)

7: ACSLS 7.3 or later

E: Exit
```

Choose the appropriate version.

WARNING: If you select an earlier ACSLS release that does not support a library, you will be prompted to remove the library from your configuration before exporting your database.

Exporting to a Disk File

You can export the ACSLS database and ACSLS control files to a disk file, as shown in the following procedure.

To export the ACSLS database and ACSLS control files to a disk file:

1. Log in as `acsss`.

2. Shut down ACSLS:

   `idle` (from cmd_proc)

   `kill.acsss` (from a UNIX command prompt)

3. Shut down the database:

   db_command stop

4. Start the db_export.sh utility.

```
db_export.sh -f /path/db_file
```

5. Select the desired option for the version to which you are migrating.

   • As it executes, the utility displays output indicating successful table data being exported.

   • When the export is complete, a message is displayed indicating that the export has been successful.

   • The `db_export.sh` utility creates two files: *db_file* and *db_file.misc* in the location specified with the -f option.

6. Ensure that these files are placed in or moved to a secure location where they won't be removed.

   Do **not** put these files in or under the following directories, because these directories may be removed or deleted when ACSLS maintenance is installed:

   • $ACS_HOME
     (the ACSSS home directory)

   • $ACSDB_BACKUP_DIR
     (directory where ACSLS backups are stored)

   • /tmp

   Note: If you plan to install a new release of the operating system, do **not** save the exported files on the ACSLS server.

Exporting to Tape

You can export the ACSLS database and ACSLS control files to tape, as shown in the following procedure.

To export the database and ACSLS control files to tape:

1. Log in as `acsss`.

2. Shut down ACSLS:

   `idle` (from cmd_proc)

   `kill.acsss` (from a UNIX command prompt)

3. Shut down the database:

   ```
   db_command stop
   ```

4. Insert a blank tape into the default tape device.

5. Start the `db_export.sh` utility;

   ```
   db_export.sh -f tape_device
   ```

Example: `db_export.sh -f /dev/rmt/0mn`

6. Select the desired option from which you are exporting.

   As it executes, the utility displays output indicating successful table data being exported and successful ACSLS files being backed up. A message displays when the export is completed.

7. Remove the cartridge from the drive only when the program completes and the prompt re-appears.

   CAUTION: You will lose files if you remove the cartridge before the program completes the export. Write protect the cartridge and clearly mark it to identify the contents as the exported database.

   Do not leave the cartridge in the library.

■ Importing the Database

The following attributes are imported into the new database when you use the `db_import.sh` utility.

- Volumes: These database tables include all of the information associated with each volume in the library, such as:

  - where the volume resides

  - type of cartridge (data, scratch, cleaning, etc.)

  - last associated scratch pool

  - current status of the cartridge (home, mounted, etc.)

  - entry date and last accessed date

  - number of mounts since the entry date

  - maximum use (for cleaning cartridges)

  - associated lock ID and user ID (if the cartridge is locked)

- ACS and Library: database tables include the ACSs and library components, such as: LSMs, drives, panels, and cells

- ACSLS control files include all configuration updates since the initial installation, including:

  - access control information

  - fixed volume preferences

  - scratch media preferences

  - custom volrpt templates

- Dynamic and static variables: dynamic variables that have been customized in a previous release can be imported.

This section describes how to use the `db_import.sh` utility to:

- recreate the ACSLS database

- recover important ACSLS control files

- recover customized dynamic variables from data exported using the `db_export.sh` utility.

Importing From a Disk File

You can import the ACSLS database and ACSLS control files from a disk file, as shown in the following procedure.

To import the ACSLS database, ACSLS control files, or customized dynamic variables from a disk file:

1. Log in as `acsss.`

2. Shut down ACSLS:

   `idle` (from cmd_proc)

   `kill.acsss` (from a UNIX command prompt)

3. Start the `db_import.sh` utility.

   `db_import.sh -f` *db_file*

   The following menu displays:

   ACSLS Import Utility

   Would you like to do:

   1) Import database only from either a different platform or from an earlier release

   2) Import customizable configuration files ($ACS_HOME/data/external only)exported from a different platform or from an earlier release

   3) Recover a previous environment on the same platform and version (Disaster Recovery) - database , $ACS_HOME/data/external and internal

   4) Merge customized dynamic variables from a previous installation

   E) Exit

   Please select one of the above:

- **Option 1 - importing only database files**

  Use this option to import the database files only. This option destroys the existing database tables, rebuilds them, and then populates them with the data provided from the exported database. Output from selecting option one is similar to the following:

```
WARNING:
This script will destroy all tables from the ACSLS database. The
results are final and there is no recovery without rebuilding
the database. If you wish to preserve information in existing
tables, you should not continue unless you have exported the
table data using db_export.sh.

You will need to rebuild the database tables using acsss_config
if you are not using db_import.sh utility.

Do you wish to continue? (y or n): y

... removing ACSLS database tables.

Creating acstable
Creating porttable
Creating lemtable
Creating captable
Creating lockidtable
Creating drivetable
Creating volumetable
Creating celltable
Creating pooltable
Creating audittable
Creating csitable
Creating paneltable
Creating vactable
Creating scr_distr_table
Creating displaycommand table
Creating displayfields table
Creating displayoptions table
Creating displayoptval table
Creating displaysubfields table
Creating ptptable
Creating clienttable
Creating handtable
Creating imutable
db_import.sh 1657: Beginning database import phase.
db_import.sh 1658: Start copy of data into empty tables.

Loading Display database reference tables.

db_import.sh 1660: Database Import Phase Complete.

ACSLS mandates backup of database in this scenario. Would you
like to make an additional backup on tape? (y or n):
```

Selecting Y initiates an extra tape backup of the database. A backup to disk is created regardless of how you answer.

- **Option 2 - recovering ACSLS control files**

  Use this option to recover ACSLS control files either from a previous version or the same version. This recovers all files in the directory acs.home under data/external including access control files. If

access control is configured, it will also recover `data/internal/client_config`. Output is similar to the following:

```
extract_misc.sh 1663: Extracting ACSLS data files.
db_import.sh 1575: Restoring ACSLS control files...
db_import.sh 1577: ACSLS control files
(/export/home/ACSSS/data/external) have been restored
```

- **Option 3 - recreating an ACSLS environment**

  Use this option to recreate an ACSLS environment. This would be used:

  - recovering from a hardware failure or during a hardware upgrade

  - when you need to rebuild the ACSLS server to be identical to the ACSLS server from which the data was exported

  This option is a combination of options 1 and 2. When finished with this option, you have the option of backing the database up to tape.

  Selecting "N" at this option, performs a local disk backup.

- **Option 4 - recovering customized dynamic variables**

  Use this option to recover customized dynamic variables from previous environments. This is a very useful option for upgrading versions of ACSLS without having to record previous customized dynamic variables.

  Note: The dynamic variable, ISOLATION_LEVEL, was removed in ACSLS 7.2.0 and is no longer used.

  If you are importing dynamic variables from a version prior to ACSLS 7.2.0, you receive a message similar to the following. You can ignore this message.

  ```
  variable <variable name> is obsolete was removed
  ```

  Selecting this option gathers the settings from the database export, and then re-configures shared memory with the new variable settings.

  WARNING: If you start ACSLS prior to executing this option, certain data could be lost. If you are upgrading ACSLS from a previous version and had customized dynamic variables, these changes must be made to your new environment PRIOR to starting ACSLS.

  Output for this option, depending on your settings, is similar to the following:

```
extract_misc.sh 1663: Extracting ACSLS data files.
File
(/export/home/ACSSS/tmp/backup/data/internal/dynamic_variables/
dv_extract) is readable
Saved current configuration into file saved_cfg_20040915_1403 in
/var/tmp/saved_cfgs/
Changing the current configuration to the previously saved
configuration. Setting (TIME_FORMAT) to %Y-%m-%d)
Change complete

Configuration change complete, no errors detected.
Please review /tmp/reset_cft.tmp7.
ACSLS dynamic variables have been successfully merged.
```

4. Verify the install as described under "Verifying the Imported Database and Library Configuration" on page 113.

Importing from Tape

Use the following procedure to import the ACSLS database, recover ACSLS control files, and rebuild customized dynamic variables from tape.

1. Log in as `acsss`.

2. Shut down ACSLS:

   `idle` (from cmd_proc)

   `kill.acsss` (from a UNIX command prompt)

3. Insert the exported database tape that you exported with the `db_export.sh` command into the tape drive.

4. Run the database import utility by entering the following at a UNIX command prompt.

   `db_import.sh`

   The `db_import.sh` utility displays its main menu as shown in "Importing From a Disk File" on page 109.

   Note: The following options are valid when importing.

   - Select option '1' and then option '2'.
   - Select option '1' and then option '3'.
   - Select option '1' and then option '4'.

   If you use any of the following combinations, you must: exit from the import utility after executing any of the options above; then restart the `db_import` utility. You can then select the second option.

   - Selecting option '2' and then option '3'.

- Selecting option '2' and then option '4'.
- Selecting option '3' and then option '2'.
- Selecting option '3' and then option '4'.
- Selecting option '4' and then option '2'.
- Selecting option '4' and then option '3'.

Note: You receive an "unsuccessful"message if you are running the `db_import` utility from one terminal and doing a rewind from a different terminal.

5. Refer to the step "Start the db_import.sh utility." on page 109 for menu options.

6. Verify the install as described in "Verifying the Imported Database and Library Configuration."

7. Import from tape, other than the default tape device (no rewind).

■ Verifying the Imported Database and Library Configuration

Use the following procedure to mount or dismount a cartridge to verify ACSLS.

Mount/dismount a cartridge to verify ACSLS:

1. Verify that you are logged in as `acsss`.

2. If ACSLS is not running, start it by entering

   `rc.acsss`

3. Query the server from the `cmd_proc` by entering

   `query server`

   If messages are displayed indicating that the server is in recovery mode, wait for a message indicating that the server is running.

4. Verify that the at least one of the following are online. If not, bring them online with the `vary` command.

   `query port all`

   `query acs all`

   `query lsm all`

   `query drive all`

5. Do you have at least one cartridge in an LSM?

YES	Continue with the procedure.
NO	Enter a cartridge into an LSM.

6. Mount a cartridge by entering:

    `mount` *`vol_id drive_id`*

    Hint: Use the query drive command to get the ID of an available drive and the query volume command to get the ID of a library cartridge. See

7. Did you see a message indicating a successful mount?

    A successful mount message is:

    `Mount:` *`vol_id`* `mounted on` *`drive_id`*

YES	Procedure is complete.
NO	If an error message appears, run this verification procedure again, ensuring that you specified a valid, available drive and a library cartridge. If the mount/dismount still fails, call Sun StorageTek for assistance.

8. Dismount the cartridge by entering:

    `dismount` *`vol_id drive_id`* `force`

    where *`vol_id`* is the volume and *`drive_id`* is the drive you specified in Step 6.

■ Automatic Database Backup

ACSLS automatically creates a backup file of the database to disk every 24 hours at midnight or the time of day and days of the week you specified in the backup options in `acsss_config`.

■ Performing Manual Backups to Tape

In addition to the automatic database backups that ACSLS creates, you should periodically run the `bdb.acsss` utility to manually create tape backups that can be stored offsite and used, if needed, for disaster recovery of the database.

Regular backups transferred to an offsite device can enable rapid restoration in the event of disaster to the ACSLS server.

Sun StorageTek recommends that you use `bdb.acsss` to manually back up the database to tape after:

*   Running `acsss_config`.

*   Importing the database.

- An audit of the entire library.

- Any database recovery.

■ Backing up to a Specified Tape Device Attached to the ACSLS Server

To back up the ACSLS database to a specified tape device attached to the ACSLS server, do the following:

1. Log in as `acsss`.

2. Insert a blank tape into the tape device.

3. From a terminal window, enter the following command:

   `bdb.acsss -f` *`tape_device`*

   Where *`tape_device`* specifies a tape device attached to the ACSLS server.

4. Messages reporting the progress of the backup appear.

   Wait for the following message to appear:

   ```
   Check tape device (/dev/rmt/0mn) to make sure you have
   a tape in the tape drive.
   [Hit RETURN to continue or Ctrl-C to exit]
   ```

   Press [[Return]].

5. Wait for the following message to appear:

   ```
   ACSLS database backup successfully completed.
   ```

   Example - To back up the ACSLS database to tape device `/dev/rmt/0mn`, enter the following command:

   bdb.acsss -f /dev/rmt/0mn

■ Backing up to a UNIX File

Hint: In the interest of disaster recovery, Sun StorageTek does not recommend that you back up to a UNIX file unless the file is on a remote disk.

Refer to .

To back up the ACSLS database to a UNIX file, do the following:

1. Log in as `acsss`.

2. From a terminal window enter the following command:

```
bdb.acsss -f db_file
```

Where *db_file* specifies a UNIX file to contain the ACSLS database. You must have write permissions to the file.

3. Wait for the following message to appear:

```
ACSLS database backup successfully completed.
```

■ Recovering and Restoring

This section describes the following restoration/recovery procedures:

- Restoring a corrupted or lost database to the most recent backup

- Restoring a corrupted or lost database to a specified date and time

- Recovering from a disk failure

- Disaster recovery for a failed server

- Recovering from a specific backup file

- Restoring non-database, ACSLS contro files

Most of these procedures use the `rdb.acsss` utility, which provides options for restoring a database from the most recent backup or from a specified date and time; disaster recovery using a backup created by `bdb.acsss`; and restoring ACSLS control files created by `bdb.acsss`. For more information about these options, see "rdb.acsss" on page 166.

Note: If the home cell of a cartridge changes from its last location after a backup, then the restored database will not be up-to-date. To avoid cartridge movement on dismounts:

- each LSM must be the only LSM in its ACS (true in most SCSI libraries), or

- the Extended Store Feature must be enabled for all LSMs that are connected to other LSMs via a pass-thru-port.

  For more information, see "Using the Extended Store Feature" on page 60. If the Extended Store Feature is not enabled for all connected LSMs or cartridges have been entered or ejected, you must audit the library after the restoration to make the database current.

Note: Do not specify the `-f` option as a general option for the `rdb.acsss` utility. If you backed up your database to an external network file or to an alternate tape device, you use the `-f` option only after entering rdb.acsss and then choosing the third recovery option. When prompted, enter `-f` and the path name to your external network file or

alternate tape device. See for more information.

■ Restoring the Database to the Most Recent Backup

In this procedure you restore the database to the most recent backup created on the local disk by automatic backups. ACSLS control files are also restored.

To restore a corrupted or lost database to the most recent backup:

1. Log in as `acsss`.

2. From the cmd_proc window, idle ACSLS:

   `idle`

3. Shut down ACSLS:

   `kill.acsss`

4. Shut down the database:

   `db_command stop`

5. Enter the following command:

   `rdb.acsss`

6. Select option 1:

   `1. Restore from a current local disk backup`

7. Refer to for procedures.

8. To start ACSLS, enter the following command:

   `rc.acsss`

■ Recovering from a Failed Server

Use this procedure for a disaster recovery when you have lost or corrupted both primary and secondary disks.

To recover from a failed server:

1. Install the operating system.

2. Install ACSLS.

   CAUTION: You *must* install ACSLS in the same directory you used before the disk failure.

3. Log in as `acsss`.

4. From the cmd_proc window idle ACSLS:

   `idle`

5. Shut down ACSLS:

   `kill.acsss`

6. Shut down the database:

   `db_command stop`

7. Enter the following command:

   `rdb.acsss`

8. Select option 2:

   `2. Restore from a previous tape or network file backup`

9. Refer to for procedures.

10. To start ACSLS, enter the following command:

    `rc.acsss`

11. You must run `acsss_config` to re-specify automated backup date and time and retention periods *unless* you want to accept the default settings.

■ Restoring ACSLS Control Files

In this procedure you restore ACSLS control files. These are non-database files that include all files in the data/external directory such as access control files, the fixed volume file, the scratch preferences file, and custom volrpt files. These files are restored from a bdb.acsss backup to tape or an external network file.

To restore ACSLS control files:

1. Log in as `acsss`.

2. From the cmd_proc window, idle ACSLS:

   `idle`

3. Shut down ACSLS:

   `kill.acsss`

4. Shut down the database:

   `db_command stop`

5. Enter the following command:

   `rdb.acsss`

6. Select option 4:

   `4. Restore only ACSLS non-database control files`

7. Refer to "rdb.acsss" on page 166 for procedures.

8. To start ACSLS, enter the following command:

   `rc.acsss`

■ Restarting the Database

Use this procedure to start up the database manually. Normally, the database is started automatically when ACSLS initializes. Typically, you would use this procedure if the database is shut down but ACSLS is still running (for example, during a recovery).

To restart the database, do the following:

1. Log in as `acsss` and open a UNIX command tool.

2. Enter the following command:

   `db_command start`

Reporting and Logging

<div style="text-align: right;">**9**</div>

In this chapter, you will learn how to:

- Create a custom volume report

- Create a report for volume movement statistics

■ Creating a Custom Volume Report

You can use the `volrpt` utility to create a volume report; for more information, see the "volrpt" on page 175. `$ACS_HOME/data/external/volrpt/owner_id.volrpt` is a sample input file that you can run or use as a template to create customized volume reports. You can also save your customized volume reports in the `$ACS_HOME/data/external/volrpt` directory. Use this procedure to create an input file to the volrpt utility to create a custom volume report.

To create a custom volume report, do the following:

1. Log in as `acsss`.

2. Open a UNIX command tool.

3. Change to the custom volume report directory:

   `cd /home/ACSSS/data/external/volrpt`

4. Copy the sample volume report file to a new, custom file.

   `cp owner_id.volrpt` *my.volrpt*

   my.volrpt is the name you want to assign to the new file.

5. Using a text editor, such as vi, edit the *my.volrpt* file to specify the fields and formats you want to see on the custom report.

- You can specify any of the fields listed in the sample file.

- The format for each entry is:
  field_name field_length delimiter_length

- You can make the field lengths and delimiters any size you wish. Just be sure all the fields you specify will fit on one line when the report prints.

- Detailed editing instructions are given in the sample file.

6. When you are finished editing the file, save it.

Custom Volume Report

You create a customized report by designating fields, field length, and delimiter lengths in an input file, such as the one shown in Figure 9:

Figure 9. Sample Volrpt Input File

```
##############################################################################
#
# File name: owner_id.volrpt
#
# This file describes the report layout for volrpt invoked with the -f option.
#      volrpt -f <filename>
#
# The format of a line is:
#      field_name         field_length              delimiter_length
#
# The field length is the number of characters which will be printed for
# the field. The delimiter length is the number of spaces that will be
# printed after the field. If you leave out the lengths, or specify a
# value of -1, the default values will be used. Default delimiters are
# always 2. Here are the fields and their default lengths.
#
# ACCESS_COUNT       5   2
# ACCESS_DATE       15   2
# CELL_ID           14   2
# DRIVE_ID          10   2
# ENTRY_DATE        15   2
# LABEL_ATTR         5   2
# LOCK_ID            5   2
# LOCK_TIME         15   2
# MAX_USE            5   2
# MEDIA_TYPE         7   2
# OWNER_ID          20   2
# POOL_ID            5   2
# VOLUME_ID          6   2
# VOL_STATUS        17   2
# VOLUME_TYPE        4   2
#
# Revision History:
# xx/xx/xx  Name      Changes
#
##############################################################################
VOLUME_ID          6      2
MEDIA_TYPE          7      2
DRIVE_ID           12      2
CELL_ID            14      2
OWNER_ID           -1      0
```

From this input file you generate the customized report shown. Among other uses, a customized report (Figure 10) allows you to report the owners of volumes.

Figure 10. Customized Volume Report Using Input File

```
1998-06-30 13:22:07

TOTAL  VOLUMES:  2        SEQUENCE:     sort by volume identifier

Volume  Media                       Home          Owner
Label  Type             Drive ID    Location      ID

RB1400 3480             Not-in-drv  0, 1, 1, 0, 0  cray
RB1401 DD3A             0, 0, 1, 0  0, 1, 2, 0, 0  cray
```

Custom Volume Report Example

The following volrpt includes ACSs 0 and 1, absent and ejected volumes, and shows the status of volumes (Figure 11 and Figure 12 on page 124).

The input volrpt options are:

```
volrpt -f my.volrpt -a 0 1 -i
```

These control statements are used to select and format the output:

Figure 11. Sample volrpt Control Statements

```
CELL_ID           14        2
VOLUME_ID         6         2
VOL_STATUS        17        0
POOL_ID           5         2
ACCESS_COUNT      5         1
LOCK_ID            5         1
OWNER_ID          20        0
```

Figure 12. Customized volrpt Showing Absent and Ejected Volumes

```
VOLUME REPORT UTILITY
2002-06-03 15:27:48

TOTAL VOLUMES: 61      SEQUENCE: sort by volume identifier

Home            Volume   Volume        Pool  Times  Lock Owner
Location---     Label    Status        ID    Mount  ID   ID-----------

1, 0, 0, 0, 0   ABC001   VOLUME_HOME     0    2      0     presc
0,-1, 0, 0, 0   ABC002   VOLUME_ABSENT   0    0      0
0,-1, 0, 0, 0   ABC003   VOLUME_ABSENT   0    0      0
1, 3, 0, 0, 3   ABC004   VOLUME_MISSING  0    0      0
1, 3, 0, 0, 4   ABC005   VOLUME_MISSING  4    0      28001 tom
1, 3, 0, 0, 5   ABC006   VOLUME_MISSING  0    0      0
0,-1, 0, 0, 0   ABC007   VOLUME_ABSENT   0    0      0
1, 0, 0, 0, 7   ABC008   VOLUME_HOME     0    0      0
0,-1, 0, 0, 0   ABC009   VOLUME_ABSENT   0    0      0
0,-1, 0, 0, 0   ABC010   VOLUME_ABSENT   0    0      0     presc
1, 0, 0, 0,10   ABC011   VOLUME_HOME     0    0      0
1, 0, 0, 0,12   ABC012   VOLUME_HOME     0    0      2371  abc012
1, 0, 0, 0,13   ABC013   VOLUME_HOME     0    0      28001
1, 0, 0, 0,14   ABC014   VOLUME_HOME     0    0      28001
0,-1, 0, 0, 0   ABC015   VOLUME_ABSENT   1    0      29712
0,-1, 0, 0, 0   ABC016   VOLUME_EJECTED  1    0      29712
0,-1, 0, 0, 0   ABC017   VOLUME_ABSENT   1    0      29712
0,-1, 0, 0, 0   ABC018   VOLUME_ABSENT   1    0      29712
1, 0, 0, 0,19   ABC019   VOLUME_HOME     1    0      0
1, 0, 0, 0,20   ABC020   VOLUME_HOME     1    0      0
0,-1, 0, 0, 0   ABC021   VOLUME_ABSENT   0    0      0
0,-1, 0, 0, 0   ABC022   VOLUME_ABSENT   4    0      0
```

■ Creating a Logging Volume Statistics Report

You can use the volume statistics log file (`acsss_stats.log`) to log volume movement statistics (Figure 13 on page 125). These statistics consist of entries for each time ACSLS detects that a volume's location changed. ACSLS logs entries for enters, ejects, mounts, dismounts, and for each time an audit detects that a volume's location has changed (typically, by being manually moved).

You use the `acsss_config` configuration program to do the following:

• Enable or disable volume statistics logging via the LIB_VOL_STATS variable

• Specify the maximum size of the volume statistics log file

• Specify the number of rollover files for the volume statistics log file

The `stats_report` utility uses the `acsss_stats.log` to report all mounts and tapte drive usage

The volume statistics log file contains collection mode entries that tell if volume statistics logging is enabled or disabled and *volume statistics entries*.

Figure 13. Example of Volume Statistics Log File Entries

```
1998-06-30 08:53:00 CONFIG
Library volume statistics on.

1998-06-30 09:23:08 EJECT
U01120 Home 0,0,1,3,5 Cap 1,0,0 Client Host Id 129.81.15.25

1998-06-30 10:36:05 ENTER
PB0444 Home 0,0,4,3,5 Cap 0,0,0 Client Host Id 129.81.15.25

1998-06-30 10:42:48 MOUNT
PB0478 Home 0,0,1,35,1 Drive 0,0,1,0 Client Host Id Local

1998-06-30 10:43:19 DISMOUNT
PB0478 Home 0,0,1,35,1 Drive 0,0,1,0 Client Host Id Local

1998-06-30 10:43:19 AUDIT
RB0478 0,0,1,35,1  STATUS_VOLUME_NOT_FOUND Client Host Id JBHUTTO

1998-06-30  10:43:19  AUDIT
PB0444  0,0,1,32,1 STATUS_VOLUME_FOUND Client Host Id JBHUTTO

1998-06-30 10:45:00 CONFIG
Library volume statistics off.

2001-01-16 09:51:07 ACSCR
OA1235 Home 0,0,5,14,14 STATUS_VOLUME_NOT_FOUND Client Host Id Local

2001-01-16 09:40:13 ACSCR
OA123A Home 0,0,5,14,15 STATUS_VOLUME_FOUND Client Host Id Local
```

In Figure 13, the collection mode entries show that statistics collection started at 8:53 a. m. on May 30, 1998 and ended at 10:45 a.m. the same day. These collection start and stop times bracket the volume statistics entries for this collection period.

The format of the volume statistics entries is:

yyyy-mm-dd hh:mm:ss command

vol_id home_loc function_loc client_host_ID

Where:

yyyy-mm-dd
　　is the year, month, and day of the entry. Four-digit year formats are supported. You specify the date format via the `acsss_config` program.

hh:mm:ss
　　is the hour, minute, and second of the entry.

command

is the ACSLS command or client request that either moved the volume or (for audits) detected that the volume was moved.

MOUNT
mount request.

DISMOUNT
dismount request.

ENTER
manual or automatic mode enter request.

EJECT
eject request.

AUDIT
audit request.

ACSMV

move request

ACSCR
Volume Recovery activity. This activity is generated automatically by ACSLS processing.

vol_id
is the volume IDs

home_loc
is the volume's home (storage cell) location.

function_loc
is the volume's location for the requests that used the volume as follows:

mount or dismount requests
location is a transport ID.

enter or eject requests
location is a CAP ID.

audit requests
specifies that an audit detected one of the following errors:

STATUS_VOLUME_FOUND
The audit found a volume in a location that did not match the location specified in the database.

STATUS_VOLUME_NOT_FOUND
The audit did not find a volume in the location specified in the database.

Volume Recovery activity

Location may be a cell ID or a transport ID, indicating that Volume Recovery detected one of the following situations:

STATUS_VOLUME_FOUND
A volume was found which was not recorded in the database, and is being added.

STATUS_VOLUME_NOT_FOUND
A volume in the database was not found in any recorded location, and is being deleted.

client_host_ID
is one of the following;

•For client application requests, the host IP address.

•For cmd_proc commands, if the environment variable LIBVOLSTATS_CMD_PROC_ID is set (ASCII characters only) in the environment of the shell that started the cmd_proc, the entry is the first 12 characters of the value of the environment variable.

•For cmd_proc commands, if the environment variable LIBVOLSTATS_CMD_PROC_ID is not set or contains non-ASCII characters, the entry is Local.

316120101

Utility Reference

<div style="text-align: right;">**10**</div>

This chapter discusses the following ACSLS utilities:

determines a list of volumes that are to be ejected and creates a file containing the set of eject commands for the specified CAP.

identifies any available port that can be used by acsss.

enables you to either enable or disable automatic startup and shutdown of ACSLS during system startup or shutdown.

allows you to monitor and manage the free cells in libraries controlled by ACSLS.

displays details about the installed ACSLS license. This includes the number of volumes and free cells currently supported by ACSLS.

filters the acsss_event log to include or exclude messages containing specific keywords.

terminates ACSLS.

moves multiple cartridges to one or more LSMs.

displays the model number, revision level, and Target-LUN address of each device connected behind an Emulex (LP10000) or QLogic (QLA2300) fibre-channel HBA.

starts ACSLS.

restores the ACSLS database and ACSLS control files.

shows detail for every mchanger device configured on Solaris.

gathers library volume statistical information.

creates a volume report.

automatically assigns ownership and pool association to volumes as they are entered through the CAP.

■ Using the ACSLS Utilities

Follow these general guidelines for using the ACSLS utilities:

- You must log in as `acsss` to ensure the proper environment to run ACSLS utilities.

  You should not `su` to `acsss`.

- Sun StorageTek recommends that you use `bdb.acsss` to manually back up the database to tape after:

  - Configuring your library hardware.

  - Importing the database. After you upgrade to a new version of ACSLS, do not use database backups created with previous versions. Make sure, however, to create a new backup as soon as you have upgraded.

  - Any database recovery.

- To ensure that you recover an accurate and consistent database, always use the most current database backup.

If a utility fails, retain all event logs. These logs will aid Sun StorageTek in resolving any problems.

■ bdb.acsss

The `bdb.acsss` utility backs up the ACSLS database contents and ACSLS control files that you need to rebuild the ACSLS environment. The backup is either placed in a tar file or tape device you have named, or in the directory defined as the default.

This utility performs ACSLS database backups without requiring ACSLS to be down (hot backup).

Without the *-f* option, a `/export/backup/<`*time_stamp*`>.tar` file is created. *Time_stamp* is the time the `bdb.acsss` command was executed.

Note: To verify the contents of the tape after running `bdb.acsss` please modify the following examples for your specific tape devices.

- For Solaris:

  `tar tvbf 2048 /dev/rmt/0mn`

- For AIX:

  `tar tvbf 20 /dev/rmt1.1`

After running the `tar tvbf` command, the tape advances to the next block. Make sure you have rewound or re-positioned the tape if you intend to run `rdb.acsss` after running the `tar tvbf` command.

Format

```
bdb.acsss -f [backup_file | tape_device]
```

If you enter `bdb.acsss` with no options, a default backup is performed that provides you the ability to restore the database back to the time when this backup was performed.

Options

`-f backup_file`
specifies a UNIX file to contain the ACSLS database backup. You must have write permissions to the file and directory.

`-f tape_device`
specifies any tape device attached and configured to the ACSLS server.

Usage

Use the `bdb.acsss` utility to back up the ACSLS database to tape or to an external network file to create backups that can be used, if needed, for recovery of the database.

It is recommended that you use bdb.acsss to manually back up the database after:

• Running `acsss_config`.

• Importing the database. After you upgrade to a new version of ACSLS, *do not* use database backups created with previous versions.

• An audit of the entire library.

• Any database recovery.

Example 1

`$ bdb.acsss -f /export/backup/my_backup`

In this example, a file named `my_backup` was created in the directory `/export/backup`. You can now choose whether to keep the file where it is or if to move it to another file system, another server, or a writable CD device.

This file can then be used to restore the database to the state it was in when the backup was performed.

For example, if the backup was performed at 1:00 pm on Friday and a restore is performed at 6:00 am on Monday, the database will revert back to its state as of 1:00 pm on Friday.

Within this same *-f* option, you can give a tape device instead of a file name and the backup will go to the named tape device.

Example 2

`$bdb.acsss -f /dev/rmt/0mn`

In this example, a tape archive on the tape device `/dev/rmt/0mn` was created. This can be stored for later use in an offsite location.

See also:

■ Dynamic Configuration (config) utilities

The dynamic configuration (config) utility allows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration changes are recorded in the `acsss_config.log` file.

The following dynamic configuration utilities are supported:

- `config acs`

- `config drives`

- `config lsm`

- `config ports`

Using the `config` utility provides the following benefits:

- ACSLS can continue running, allowing you to perform mount requests to unaffected library components.

- Allows you to reconfigure specified library components while all other configuration information remains unchanged. For example, when specifying:

  - An ACS, the configurations of other ACSs are not affected.

  - An LSM, the configurations of other LSMs are not affected.

  - A drive panel, the drives on a panel, mounts and dismounts to all existing drives are not affected.

Important

- ACSLS must be running to use the dynamic `config` utility.

- You will need to enter a new licence key if you have increased the number of storage cells in your existing configuration and that increase now exceeds your current licensed capacity.

You will then need to perform an audit of the library. This will update the ACSLS database to match the actual inventory of library cartridges.

- You must use `acsss_config` to create your initial ACSLS configuration. Refer to "Setting Variables that Control ACSLS Behavior" on page 31.

- Event notification reports all dynamic configuration changes.

- Before running dynamic configuration, ensure that all components being added or re configured are ready.

- The `acsss_config.log` file provides details regarding messages displayed.

- If you have not confirmed the configuration change, you can cancel the operation with [[CTRL]]+C.

- Dynamic configuration performs an automatic backup before and after the configuration change.

- After a configuration change is confirmed, it cannot be cancelled. If you want to reverse a configuration change, you should shut down ACSLS and restore the backup that was taken immediately before the configuration change was made.

  You have 10 minutes to confirm a configuration change before it times out.

- You cannot remove the only (or last) ACS.

Limitations of Dynamic Configuration

The dynamic configuration utility has two significant limitations:

- You cannot delete an ACS, or delete or change a port (connection) to a library.

- With an existing SCSI/fibre-attached library, you can **only update** drive configurations. The `config drives` utility works without limitations. The `config acs` and `config lsm` utility will only update the drives in a SCSI/fibre-attached library. You cannot update panel cell or CAP configurations in a SCSI/fibre-attached library. `config acs new` works for SCSI/fibre-attached libraries.

Solution:

For these configuration changes that are not supported through dynamic configuration, you need to shutdown ACSLS and use `acsss_config`.

Things You Should Not Do

- Do **not** use dynamic configuration to display status information for a library and its components because it involves extensive I/O to the library

  Use the `query` or `display` commands instead.

- Try to perform more than one configuration task at a time.

  Only one dynamic configuration task can be performed at a time. This:

  - Minimizes performance problems caused by the I/O between ACSLS and the library it is configuring,

  - Avoids complex interactions between multiple configuration tasks.

■ config acs

The `config acs` utility allows you to:

- Add an ACS or re-configure an existing ACS and its components

- You can configure or re-configure libraries without assigning all ACS numbers in sequence.

  Example: You want to migrate from a 9310 library to a SL8500 and then remove the 9310. The 9310 ACS is numbered ACS 0 and the SL8500 ACS is ACS 1. Using this feature, you can now migrate all of the cartridges and drives to the SL8500 ACS and remove the 9310 ACS without having to renumber your SL8500 ACS.

- Configure a partitioned ACS

- Add or remove LSMs, since the `config lsm` utility only allows you to reconfigure an existing LSM.

  WARNING: All ACSs must have at least one CAP. This can be a shared CAP.

Format

To add a new ACS, enter:

    config acs new

To reconfigure an existing ACS, enter:

    config acs *acs_id*

Adding a new ACS

To add a new ACS:

1. Enter:

    ```
    config acs new
    ```

2. Specify the ACS number for the new ACS.

    ACSLS 7.3 lets you configure or re-configure libraries without assigning all ACS numbers in sequence.

    The ACS numbers alreay used and the first five available ACS numbers are displayed.

3. Enter the ACS number for your new ACS.

4. Select whether the ACS is in a partitioned library (SL8500).

    - If you enter **y**, you are asked for the partition ID for the ACS.

      This partition ID must match the partition ID on the SLConsole.

    - If this is not a partitioned library, enter **n.**

5. Enter the number of connections to the ACS followed by the device or host ID(s).

    You can have a maximum of four connections.

    CAUTION: **Ensure that all ports are connected to the same ACS.**

    The new ACS configuration is displayed.

6. Confirm the addition of the new ACS.

    After confirmation, the configuration information is displayed and the database is updated.

Reconfiguring an existing ACS

The ACS should, if possible, be online or in diagnostic mode when you reconfigure the ACS.

To reconfigure the ACS:

1. Enter:

    ```
    config acs acs_id
    ```

    The old and new configurations are displayed.

2. Confirm the new configuration.

    After confirmation, the database is updated.

    If the configuration is unchanged, the configuration is displayed without asking for confirmation, and the utility exits. Examples of this are:

- Only drive types and/or serial numbers change

- LSM serial number changes

- Only LSM type changes between 4410 and 9310

- Number of hands, such as SL8500 robots, change.

However, if these changes occur with other changes requiring confirmation, then you must confirm the new configuration. The database is then updated.

The ACS and its components are removed from the database

Limitation of config acs

The `config acs` utility will only update drive configurations in a SCSI/fibre-attached library.

It will not update panel or CAP configurations. In order to update them, you must shutdown ACSLS and use `acsss_config`.

■ config drives

The `config drives` utility allows you to reconfigure all drives on an existing drive panel. This includes, adding drives, updating drive types and serial numbers for existing drives, and deleting drives that were removed from the database.

Use the `config drives` utility for dynamic changes to drive configurations, which include installation, replacement, or removal of drives on an existing drive panel. Other changes to tape library hardware configurations such as changes in the number and/or location of storage cells, number or size of CAPs, or replacement of a storage cell panel with a drive panel must be made using the `config lsm` or `config acs` utility. Please note:

- The LSM containing the panel with the changed drive configuration must be online or in diagnostic mode.

- On the drive panel being re-configured, all drives must be ready.

- On a SCSI-attached library, all drives must have been ready when the library was last IPLed, and the drives must still be ready.

- When new drives replace existing drives, varying the LSMs or drive types online or running `config drives` automatically updates the drive types and drive serial numbers.

Format

    config drive *panel_id* or config drives *panel_id*

Usage

To reconfigure all drives on an existing drive panel:

1. Enter:

   `config drive` *panel_id* or `config drives` *panel_id*

   The old and new drive configurations are displayed for the panel.

2. Confirm the configuration change.

   After confirmation the database is updated.

   - If the configuration has not changed, the configuration is displayed without asking for confirmation, and the utility exits.

   - If only drive types or serial numbers changed, the ACSLS database is updated without requesting confirmation.

■ config lsm

This utility allows you to reconfigure an existing LSM and all of its components. These components include CAPs, panels, and drives.

If you want to add or delete an LSM in an ACS, you must use the `config acs` utility.

Procedures for when Panels Change:

- If there are panels that are being removed or changed and have been emptied of cartridges, the LSM can remain online.

- If there are panels that are being removed or changed and contain cartridges, it is recommended that you vary the affected LSM to diagnostic until you re-configure the LSM and audit the panels affected. If you do not do this, mounts and dismounts may fail.

- If you have added panels and have manually placed cartridges in these panels, please run an audit to reconcile the database.

Format

`config lsm` *lsm_id*

Usage

To reconfigure the LSM:

1. Enter:

   `config lsm` *lsm_id*

The old and new configurations are displayed.

A "y" next to the panel notifies you that the panel type(s) have changed. Look in the `acsss_config.log` file for details.

2. Confirm the new configuration.

After confirmation, the database is updated.

If the configuration is unchanged, the configuration is displayed without asking for confirmation, and the utility exits.

Minor changes are made automatically without confirmation. Examples are:

- Only drive types and/or serial numbers change

- LSM serial number changes

- Only LSM type changes between the 4410 and 9310

- Number of hands, such as SL8500 robots, change.

However, if these changes occur with other changes requiring confirmation, then you must confirm the new configuration. The database is then updated.

Limitation of config lsm

The `config lsm` utility will only update drive configurations in a SCSI/fibre-attached library.

It will not update panel or CAP configurations. In order to update them, you must shutdown ACSLS and use `acsss_config.`

■ config ports

The `config ports` utility allows you to dynamically add port connections to an ACS.

CAUTION: All new ports must be connected to the same ACS as existing ports.

run `config acs acs_id` and then `config ports acs_id.`

Important: If you want to replace one ACS with another ACS or change port connection addresses, you must shutdown ACSLS and use `acsss_config.`

Format

`config ports acs_id` or `config port acs_id`

Usage

To add ports:

1. Enter:

   `config port` *acs_id* or `config ports` *acs_id*

   The current port connection for the specified ACS are displayed.

2. Enter the number of port connections for the specified ACS.

   You can have a maximum of four connections.

3. Specify the device or host ID (s).

   Note: Ensure that new ports are connected to the same ACS as existing ports.

   The old and new configurations are displayed.

   A change in the order of the ports is not a configuration change. The connections are displayed on the order they are currently recorded in the database.

4. Confirm the new configuration.

   After confirmation, the database is updated.

   If the configuration is unchanged, the configuration is displayed, and the utility exits.

Limitations of config ports

The `config ports` *acs_id* utility is not supported on a SCSI-attached library.

The `config ports` utility will not delete or change a port (connection) to a library. In order to do this, you must shutdown ACSLS and use `acsss_config.`

■ db_command

The `db_command` command allows you to start or shut down the database, check its running status, as well as change database logging levels without interruption to ACSLS and its database.

CAUTION: You must terminate ACSLS using the `idle` command and the `kill.acsss` utility before you stop the database.

Format

```
db_command
[start|stop|stop_force|status|log_normal|log_verbose|log_
level]
```

Menu Options

Figure 14. db_command Menu Options:

```
$ db_command

    where command is one of:
      start       Start the ACSLS database

      stop        Gently bring down the ACSLS database (wait
                  for ACSLS to disconnect)

      stop_force  Quickly bring down the ACSLS database (does
                  not wait for ACSLS to disconnect)

      status      Display whether database is running or not

      log_normal  Log errors on ACSLS database activity

      log_verbose Log details about ACSLS database activity

      log_level   Display current database logging level


```

Options

db_command start
   starts the database.

db_command stop
   waits for ACSLS to disconnect, then brings down the database.

   Note: This is the preferred method to shutting down ACSLS.

db_command stop_force
   does not wait for ACSLS to disconnect before it brings down the database.

   Note: This is not a method that should be used to shut down the
             database. You should only use this method when instructed by Sun
             Support when they are troubleshooting a problem with ACSLS.

db_command status
   displays if the database is running or not running.

db_command log_normal
   logs errors on the ACSLS database activity.

db_command log_verbose
   logs details on the ACSLS database activity.

db_command log_level
> displays the current database logging level. (normal or verbose) without looking at the postgresql.conf file

■ db_export.sh

The db_export.sh utility exports the ACSLS database table data and ACSLS control files in preparation for an upgrade installation or a reinstallation of ACSLS.

Format

db_export.sh -f [*db_file* | *tape_device*]

Options

-f *db_file*
> specifies a UNIX file to contain a backup of the ACSLS database. You must have write permissions to both the file and directory.

Notes:

- If you export the database to a file, the file must reside in a non-volatile directory. If your intention is to re-install ACSLS, the re-installation will destroy the $ACS_HOME directory. Consequently, you should place the exported file elsewhere in your file system.

- If you specify the filename without a path, db_export saves the database file under that filename in the current working directory, and it saves the misc file in the $ACS_HOME directory

-f *tape_device*
> specifies any tape device attached and configured to the ACSLS server.

If no options are specified, the system defaults to the tape device.

Usage

Use the db_export.sh utility to prepare for a reinstallation of ACSLS or an upgrade installation of ACSLS.

Examples

Refer to Chapter 8, "Database Administration" for examples.

See also:

- "db_import.sh"

- "rdb.acsss" on page 166

- "Exporting the Database" on page 105

■ db_import.sh

The db_import.sh utility imports the ACSLS database table data and the ACSLS control files that you exported when you used the db_export.sh utility. The ACSLS control files are located in $ACS_HOME/data/external and consist of user definable variables and configuration for ACSLS. They specify Access Control settings, scratch preferences, Extended Store LSMs, custom volrpt settings, volume attributes (for watch_vols utility), etc. The db_import.sh utility also provides disaster recovery capabilities, as well as the retention of your customized dynamic variables when moving to a different operating system or from an earlier release.

Format

db_import.sh -f [*db_file* | *tape_device*]

Options

-f *db_file*
   specifies a UNIX file created by db_export.sh.

-f *tape_device*
   specifies any tape device attached and configured to the ACSLS server.

If no options are specified, the system defaults to the tape device.

Usage

Use the db_import.sh utility to import the ACSLS database that you exported using the db_export.sh utility.

CAUTION: The db_import utility will not run if ACSLS is running.

See also

- "db_export.sh" on page 142
- "rdb.acsss" on page 166
- "Importing the Database" on page 108

■ del_vol

The del_vol utility deletes from the database volumes that either:

- cannot be found in the library, or
- are marked as absent or ejected, or
- are in an offline LSM or drive (missing).

Note: This utility does not delete a volume that is found in the library.

You can use the del_vol utility to remove a volume record without waiting for the expiration of an absent or ejected status.

Format

del_vol [-n] [-d] [-q] *vol_id*

Options

-n

Optional; no-confirm mode; delete volume without prompting user if all locations can not be examined.

-q

Optional; quiet mode; do not print out all information extracted from the database.

-d

Optional; delete the volume; do not mark it absent. Use this option to remove absent or ejected volumes from the database.

vol_id

The volume serial number to be deleted.

The Manual Volume Delete (del_vol) utility can now retain volumes as absent. Using the -d option deletes the specified volume without waiting for the expiration of an absent or ejected status.

Note: If all of the referenced locations for the volume can't be verified (i.e., the LSM is offline or the drive is not ready), you are prompted to confirm the deletion unless the -n (no_confirm_flag) is on. After confirmation, or if the no_confirm_flag is on, the volume and information associated with it, such as scratch pool membership and current and pending locks, are removed from the database. Volume-related information is displayed unless the -q (quiet_flag) has been entered. If multiple options are used, they can be formatted either as separate options or as a contiguous string.

Usage

ACSLS and the database must be up and running (*not* idle) to use this utility. Typically, you use the del_vol utility to delete a volume from an offline LSM so that you can re-enter the cartridge into the CAP of an online LSM. Make sure that you manually remove the same volume as the volume you deleted from the database. If you delete a volume from the database, but mistakenly remove another cartridge from the LSM, return the cartridge to its proper cell in the LSM and then remove the correct cartridge.

You can use del_vol to remove a cartridge from an offline LSM and then reenter it in an online LSM, so it can be automatically mounted. To do this:

• Remove the cartridge from the offline LSM.

- Use `del_vol` to mark the cartridge as absent.

- Enter the cartridge into the online LSM.

Using `del_vol`, removes from the database the cartridge and all information associated with it, such as scratch pool membership and current and pending locks (provided the absent volume retention period is zero). All available information related to the cartridge is printed out and verified. You must manually remove the cartridge from the LSM.

CAUTION: If you mistakenly delete a cartridge from the database, you should `audit` the sub-panel containing the home cell of the deleted cartridge to re-enter the cartridge into the database.

Running `del_vol` while the system is in recovery can produce unpredictable results. The recovery sequence happens during a `vary LSM online.`

Example

To delete cartridge U01102 without a printout of the cartridge information:

del_vol -q U01102

See Also

- "Using the Manual Volume Delete Utility" on page 75

■ drives_media.sh

This routine displays all drive types, media types and drive-to-media compatibility that are supported by the current release of ACSLS. The information is normally displayed to standard output.

Format

```
drives_media.sh [ -f, -h ]
```

Options

`-f`
    the information is written to three files:

- /tmp/drive_types.txt
- /tmp/media_types.txt
- /tmp/media_compatibility.txt.

`-h`
    displays the syntax message.

Example

```
$ drives_media.sh
```

```
#
# Tape Media Supported by ACSLS
#
# ACSLS uses "Media Name" when reporting cartridge media type in
# cmd_proc, display and event log messages.
#
# "Library Media Domain" and "Library Media Type" are the values passed
# from the library hardware to ACSLS.
#
# The "ACSLS Media Type Number" is used in the ACSAPI to report the media
# type to ACSAPI clients. (This number is also used internally in ACSLS.)
#
# In the "Cleaning Cartridge?" field, three values may be defined:
#   Yes   - This media type is used exclusively for cleaning.
#   No    - This media type is never used for cleaning.
#   Maybe - This media type may apply to a data or a cleaning cartridge.
#
# Media type 24 is reserved for VSM virtual tape media.
# Virtual tapes are supported by LibraryStation, which uses media type
# 24 for virtual media in its ACSAPI communication with clients.
# To avoid conflicts, ACSLS must NOT use media type 24.
#
                                    ACSLS
Media                     Library   Media                     Max
Type        Media         Media     Type      Cleaning        Cleaning
Name        Domain        Type      Number    Cartridge?      Usage
----        ------        ----      ------    ----------      -----
3480        00            1         0         Maybe           100
3490E       00            E         1         No              0
DD3A        00            A         2         No              0
DD3B        00            B         3         No              0
DD3C        00            C         4         No              0
DD3D        00            D         5         Yes             50
DLTIII      01            C         6         Maybe           20
DLTIIIXT    01            E         8         No              0
DLTIV       01            D         7         No              0
EECART      00            Z         11        No              0
JLABEL      00            J         12        Maybe           100
KLABEL      00            K         15        Maybe           100
LTO-100G    4C            1         16        No              0
LTO-10GB    4C            C         19        No              0
LTO-200G    4C            2         26        No              0
LTO-35GB    4C            B         18        No              0
LTO-50GB    4C            A         17        No              0
LTO-CLN1    43            1         22        Yes             15
LTO-CLN2    43            2         20        Yes             50
LTO-CLN3    43            3         21        Yes             50
LTO-CLNU    43            U         25        Yes             50
SDLT        01            S         23        Maybe           20
SDLT-2      01            2         27        No              0
STK1R       00            R         9         No              0
STK1U       00            U         10        Yes             100
STK2P       00            P         13        No              0
STK2W       00            W         14        Yes             100
VIRTUAL     00            v         24        No              0


# Drive and Media Compatibility
#
# This shows the media types that are compatible with each drive type.
#
Drive Name                      Compatible Media Name
----------                      ---------------------
4480                            3480
4490                            3480
4490                            3490E
4490                            JLABEL
4490                            KLABEL
9490                            3480
```

9490	3490E
9490	JLABEL
9490	KLABEL
SD3	DD3A
SD3	DD3B
SD3	DD3C
SD3	DD3D
SD3	JLABEL
SD3	KLABEL
4890	3480
4890	3490E
DLT2000	DLTIII
DLT2000XT	DLTIII
DLT2000XT	DLTIIIXT
DLT4000	DLTIII
DLT4000	DLTIIIXT
DLT4000	DLTIV
DLT7000	DLTIII
DLT7000	DLTIIIXT
DLT7000	DLTIV
9840	STK1R
9840	STK1U
9840-3590	STK1R
9840-3590	STK1U
9491	EECART
9491	3490E
9491	3480
9491	JLABEL
9491	KLABEL
DLT8000	DLTIII
DLT8000	DLTIIIXT
DLT8000	DLTIV
T9940A	STK2P
T9940A	STK2W
9940-3590	STK2P
9940-3590	STK2W
T9840B	STK1R
T9840B	STK1U
T9840B35	STK1R
T9840B35	STK1U
HP-LTO	LTO-100G
HP-LTO	LTO-50GB
HP-LTO	LTO-35GB
HP-LTO	LTO-10GB
HP-LTO	LTO-CLN1
HP-LTO	LTO-CLNU
IBM-LTO	LTO-100G
IBM-LTO	LTO-50GB
IBM-LTO	LTO-35GB
IBM-LTO	LTO-10GB
IBM-LTO	LTO-CLN2
IBM-LTO	LTO-CLNU
SGT-LTO	LTO-100G
SGT-LTO	LTO-50GB
SGT-LTO	LTO-35GB
SGT-LTO	LTO-10GB
SGT-LTO	LTO-CLN3
SGT-LTO	LTO-CLNU
SDLT	SDLT
SDLT	DLTIV
SDLT-320	SDLT
SDLT-320	DLTIV
T9940B	STK2P
T9940B	STK2W
T9940B35	STK2P
T9940B35	STK2W
VIRT-DRV	VIRTUAL
T9840C	STK1R
T9840C	STK1U
T9840C35	STK1R

```
T9840C35                        STK1U
HP-LTO-2                        LTO-200G
HP-LTO-2                        LTO-100G
HP-LTO-2                        LTO-50GB
HP-LTO-2                        LTO-35GB
HP-LTO-2                        LTO-10GB
HP-LTO-2                        LTO-CLN1
HP-LTO-2                        LTO-CLNU
IBM-LTO-2                       LTO-200G
IBM-LTO-2                       LTO-100G
IBM-LTO-2                       LTO-50GB
IBM-LTO-2                       LTO-35GB
IBM-LTO-2                       LTO-10GB
IBM-LTO-2                       LTO-CLN2
IBM-LTO-2                       LTO-CLNU
SGT-LTO-2                       LTO-200G
SGT-LTO-2                       LTO-100G
SGT-LTO-2                       LTO-50GB
SGT-LTO-2                       LTO-35GB
SGT-LTO-2                       LTO-10GB
SGT-LTO-2                       LTO-CLN3
SGT-LTO-2                       LTO-CLNU
SDLT-600                        SDLT
SDLT-600                        SDLT-2



# Tape Drives Supported by ACSLS
#
# "Drive Name" is used to report the drive type in cmd_proc, display and
#  and event log messages.
#
# "ACSLS Drive Number" specifies the drive type in ACSAPI communications
# with clients.  (It is also used internally by ACSLS to identify the
# drive type.)
#
# Note: Drive type 23 is reserved for VSM virtual drives
#       (not supported by ACSLS).
#
                                                            ACSLS Drive
Drive Name  Drive Description                               Type Number
----------  -----------------                               -----------
4480        18 track                                        0
4490        Silverton 36 track                              1
9490        Timberline 36 track High Perf                   2
SD3         Redwood helical                                 3
4890        Twin Peaks - 36 track                           4
DLT2000     Quantum DLT2000                                 5
DLT2000XT   Quantum DLT2000XT                               6
DLT4000     Quantum DLT4000                                 7
DLT7000     Quantum DLT7000                                 8
9840        9840                                            9
9491        Timberline EE - 36 track                        10
DLT8000     Quantum DLT8000                                 11
9840-3590   IBM 3590 emulation (9840)                       12
T9940A      T9940A with 3490E image                         13
9940-3590   T9940A with 3590 image                          14
SDLT        Super DLT 220                                   15
T9840B      High Perf 9840 with 3490E image                 16
T9840B35    T9840B with 3590 image                          17
HP-LTO      HP LTO                                          18
IBM-LTO     IBM LTO                                         19
SGT-LTO     Seagate LTO                                     20
T9940B      T9940B with SCSI or 3490E image                 21
T9940B35    T9940B with 3590 image                          22
VIRT-DRV    VSM virtual drive                               23
SDLT-320    Super DLT 320                                   24
T9840C      T9840C with Native or 3490E image               25
T9840C35    T9840C with 3590 image                          26
HP-LTO-2    HP LTO Generation 2                             27
```

```
IBM-LTO-2   IBM LTO Generation 2                               28
SGT-LTO-2   Seagate LTO Generation 2                           29
SDLT-600    SDLT-600                                           30

$
```

■ ejecting.sh

The `ejecting.sh` utility takes a list of cartridges intended for ejecting and creates a file containing a set of eject commands optimized for the specified CAP. The input file must contain a simple list of cartridges that are to be ejected. The resulting series of optimized eject commands must be input to `cmd_proc` to eject the volumes. The eject commands from `ejecting.sh` are sent to standard out and can be piped directly to cmd_proc

Format

```
ejecting.sh <CAP-id> <volume list>
```

Options

`-CAP-id`
  the CAP containing the cartridges to be ejected.

`-volume list`
  list of cartridges to be ejected.

Note: The volume IDs must follow these rules:

• One cartridge-id per line.

• The vol_ids must be valid ACSLS volume IDs.

Usage

• This script checks for the validity of the CAP-id argument and if the specified CAP does not exist, it displays the message "CAP <CAP-id> not found".

• The script also determines the number of cells in the specified CAP and then formats an eject command to completely fill the CAP. The script continues to create eject commands to completely fill that cap. This is repeated until all of the cartridges in the incoming cartridge list have been included in the output.

Examples

Example 1:

In this example a series of optimized eject commands are sent to standard out and piped directly to `cmd_proc`:

```
ejecting.sh 0,0,0 volume_list_file | cmd_proc -lq
```

Example 2:

The output is redirected to a file by one of the following methods:

ejecting.sh 0,0,0 volume_list_file > eject.dat

```
ejecting.sh 0,0,0 volume_list_file | tee eject.dat
```

> **Note:** The "tee" directs the output to both a terminal screen and the specified file.

Note: If the output of this script has been redirected to a file, you may use that file as input for cmd_proc as follows:

```
cmd_proc -lq < eject.dat
```

■ find_tty.sh

The `find_tty.sh` utility assists you in finding an appropriate port to designate for a serial-attached library. It identifies available ports that can be used by ACSLS user, 'acsss'. This utility reviews all defined ports and verifies which ports are actually available. All available ports are displayed and the output reveals which of these have read-write permissions for use by 'acsss'. In addition, other available ports which are not currently accessible to user 'acsss' are displayed.

The method to identify available ports varies from one system to the next. For Solaris, the device nodes for all ports defined in `/dev/term` are displayed. For AIX, the method is a "lsdev" query.

Format

```
find_tty.sh
```

Options

Usage

- This option allows you to select an appropriate device on your system. You select a device when following prompt is displayed:

    ```
    Connect the LMU to an appropriate serial device
    ```

- If you do not know which device to use, run `find_tty.sh` to identify all eligible devices.

Example

The following example is of an output from a Solaris box with multiple ports.

```
$ find_tty.sh

/dev/term/0 is accessible to acsss
/dev/term/1 is accessible to acsss
/dev/term/2 is accessible to acsss
/dev/term/3 is accessible to acsss
/dev/term/4 is accessible to acsss
/dev/term/5 is accessible to acsss
/dev/term/6 is accessible to acsss


/dev/term/0 is not accessible to acsss.

You must set permissions on the device node:

crwx------   root   /devices/sbus@1f,0/SUNW, spif@1,0:7

/dev/term/a is accessible to acsss
/dev/term/b is accessible to acsss
```

■ fix_rc.sh

This utility enables you to either enable or disable automatic startup and shutdown of ACSLS during system startup or shutdown. Although you are presented with the option to allow automatic restart of ACSLS during installation, you may later choose to enable or disable this option. The command `fix_rc.sh` is located in the install directory of ACSLS, usually located in /export/home/ACSSS.

Format

```
fix_rc.sh
```

Options

Usage

- To install `fix_rc.sh`:

  As "root":

  ```
  # /export/home/ACSSS/install/fix_rc.sh
  ```

  ```
  Your system is currently configured in such a manner that
  when it reboots, it will *NOT* automatically restart the
  ACSLS.
  ```

```
Do you want your system configured so that when it reboots it
automatically restarts the ACSLS? (y or n): y
```

```
fix_rc.sh 1864:  Adding commands to automatically restart
ACSLS on reboot.
```

This creates the necessary files in `/etc/init.d` to allow the system init process to start and stop ACSLS at system startup and shutdown.

* To remove `fix_rc.sh`:

As "root":

```
# /export/home/ACSSS/install/fix_rc.sh
```

```
Your system is currently configured in such a manner that
when it reboots, it will automatically restart the ACSLS.
```

```
Do you want your system configured so that when it reboots it
automatically restarts the ACSLS? (y or n): n
```

```
fix_rc.sh 1865:  Removing commands to automatically restart
ACSLS on reboot.
```

This prevents ACSLS from changing states at system start and shutdown.

It is highly recommended that you allow the system init process handle startup and shutdown of ACSLS. This prevents errors caused when ACSLS is not properly brought down prior to system shutdown.

Note: If you choose not to automatically restart ACSCLS upon a reboot, then 'not classified' messages similar to the following are seen in the `acsss_event.log` after the server is rebooted:

```
2007-03-27 13:37:12 storage server[0]:
1328 N di_pri_get_status_code.c Unknown 188
di_pri_get_status_code:
 DBMS error.  Return code (28000) and message ("[unixODBC]")
not mappable to DI_STATUS
```

```
2007-03-27 13:37:12 storage server[0]:
1328 N di_pri_get_status_code.c Unknown 188
di_pri_get_status_code:
 DBMS error.  Return code (28000) and message
("[unixODBC]Communication closed during authentication;
```

■ free_cells.sh

The `free_cells.sh` utility allows you to monitor and manage the free cells in libraries managed by ACSLS. This utility reports the free cell count for LSM, ACS, and the ACSLS server.

Note: Since `free_cells.sh` reads every cell record in the database, it is recommended that you run this utility infrequently, such as only during periods of low database activity.

This utility is located in the `$ACS_HOME/diag/bin` directory

Format

```
free_cells.sh
```

Option

`-a`

displays the free cells, allocated cells and the total number of cells in each ACS and LSM, as well as those managed by the ACSLS server.

Examples

Free cells in each LSM

- LSM is not a L5500 (lists only the total free cells)

  - No options selected

    ```
    LSM 1,3

        Total free cells     =  2,345
    ```

  - -a option (also lists allocated cells and total cell count)

    ```
    LSM 1,3

        Total free cells     = 3,345
        Allocated cells      = 3,155
        Total cells          = 6,500
    ```

- LSM is a L5500 (lists LTO and non-LTO free cells separately)

  - No options selected

    ```
    LSM 0,2

        LTO free cells       =    573
        non-LTO free cells   =    467
        Total free cells     =  1,040
    ```

  - -a option (also lists allocated cells and total cell count)

    ```
    LSM 0,2

        LTO free cells       =    573
        non-LTO free cells   =    467
        Total free cells     =  1,040
        Allocated cells      =  4,460
        Total cells          =  5,500
    ```

Free cells in each ACS

If there is only one LSM in the ACS, only the ACS is listed as shown in the following examples.

- ACS does not include any L5500s

  - No options selected (displays only the total free cell count)

    ```
    ACS 1

        Total free cells      =    5,342
    ```

  - -a option (also displays allocated cells and total cell count)

    ```
    ACS 1

        Total free cells      =    5,342
        Allocated cells       =    5,658
        Total cells           =   11,000
    ```

- ACS includes L5500 (displays LTO and non-LTO free cells separately)

  - No options selected

    ```
    ACS 0

        LTO free cells        = 1,573
        non-LTO free cells    =   968
        Total free cells      = 2,541
    ```

  - -a option (also displays allocated cells and total cell count)

    ```
    ACS 0

        LTO free cells        =  1, 573
        non-LTO free cells    =     968
        Total free cells      =   2,541
        Allocated cells       =   2,959
        Total cells           =   5,500
    ```

Free cells managed by the ACSLS server

- ACS does not include any L5500s

  In this case only the total free cell count is displayed

  - No options selected

    ```
    ACSLS Server

        Total free cells      = 7,883
    ```

  - -a option

    ```
    ACSLS Server

        Total free cells      = 7,883
        Allocated cells       =14,117
        Total cells           =22,000
    ```

- ACS includes L5500s

  In the case where there are L5500 libraries being managed, only the LTO free cell and total free cell count is displayed. Non-LTO free cells are not

displayed since the L5500 library is the only library that separates LTO from non-LTO cells.

- No options selected

```
ACSLS Server

    LTO free cells        = 1,573
    Total free cells      = 7,883
```

- -a option

```
ACSLS Server

    LTO free cells        = 1,573
    Total free cells      = 7,883
    Allocated cells       =14,117
    Total cells           =22,000
```

■ get_license_info

The `get_license_info` utility displays the following information about the license:

- Customer

- Site

- Product

- Key type

- Expiration date

- Days before license expires

- Licensed cells (volumes)

- License key

- Used cells (volumes)

- Free cells (volumes)

- Total cells (volumes)

Format

`get_license_info`

Options

Example

```
$ get_license_info
Customer        STK_HQ_SUPPORT
Site             6111
Product         ACSLS700
Key Type        Permanent
Expires         2010-12-31
Days to Expire  2430
Licensed Cells  600
License Key     WIRLJP7F2YBIVIQ8KMS
Used Cells      7
Free Cells      425
Total Cells     432
```

■ greplog

Use the `greplog` utility to filter the acsss_event log to include or to exclude messages containing specific keywords. The syntax of this routine is similar to the Unix 'grep' function. `greplog` is specifically designed for use with the `acsss_event.log`, but it may function with any type of message file where the records are separated by an empty line.

Format

```
greplog -[v|i] <keyword> <logfile>
```

Options

-v

Optional. `greplog` displays all the messages in the log except those which include the keyword.

-i

Optional. `greplog` ignores the characters in the specified keyword.

-keyword

returns the complete multi-line message containing the keyword.

-logfile

list of log files.

Usage

Since the utility is specifically designed for log files, `greplog` returns the complete multi-line message containing the keyword rather than a single line

containing that word. Using the `-i` option, `greplog` ignores the characters in a specified keyword. Using the `-v` option, greplog displays all of the messages in the log except those which include the keyword. Greplog is specifically designed for use with the `acsss_event.log`, but it may function with any type of message file where the records are separated by an empty line.

■ kill.acsss

The `kill.acsss` utility terminates ACSLS.

CAUTION: ACSLS generates an error in the event log if you reboot the ACSLS server without shutting it down.

Refer to "fix_rc.sh" on page 151.

Format

`kill.acsss`

Options

None.

Usage

Use the `kill.acsss` utility to terminate ACSLS.

See also

- "rc.acsss" on page 165

- "idle" on page 215

CAUTION: Do not run `kill.acsss` while an ACSLS request is in progress! You must first idle ACSLS (with the `idle` command) before you run `kill.acsss`.

Note: `kill.acsss` does *not* shut down the database. You must issue `db_command stop` if you want to shut down the database after you shut down ACSLS.

■ moving.sh

The `moving.sh` utility moves multiple cartridges to one or more LSMs. This utility reads a file that lists the cartridges to be moved. These cartridges can be:

- Cartridges in one or more LSMs

- Cartridges on a panel being moved to other panels in the same LSM or other LSMs

- Any group of cartridges you select

The limitations of `moving.sh` are:

- All destination LSMs and cartridges in the *vol_list_file* must be in same ACS

- If any destination LSM is offline or does not contain any free cells, no cartridges are moved to that LSM

Notes:

- The `moving.sh` utility runs only if ACSLS is running

- Internally, `moving.sh` moves only one cartridge at a time to avoid impacting library performance (mounts and dismounts)

- You can run multiple move utilities in parallel after creating separate lists of volumes. Please note that if:

  - the destination LSM is same, you should make sure that there are enough free cells in the LSM to accommodate all cartridges

  - you are moving within one SL8500 library - there are only two elevators, so running more than two move utilities at a time will not increase performance

Format

```
moving.sh -f vol_list_file -t lsm_id...
```

Where:

- -f *vol_list_file*

  The file containing the list of volumes to be moved.

  Note: The volume IDs must follow these rules:

  - One cartridge-id per line.
  - The vol_ids must be valid ACSLS volume IDs.
  - If the vol_ids include trailing or leading spaces, they must be enclosed within single quotes or double quotes.

- -t *lsm_id*

  Specifies one or more LSM IDs to which the cartridges will be moved. Each LSM ID should be separated by a space

Usage

Use the `moving.sh` utility to move a list of cartridges to other LSM(s) or from one panel to other panels in the same LSM.

You can use either a custom volume report or the `display volume` command to create a file containing the list of volumes to be moved from an LSM.

You want to use the `moving.sh` utility:

- When a SL8500 is initially partitioned or repartitioned, and one or more rails (LSMs) are removed from an existing partition (ACS), `moving.sh` can move cartridges from the LSM being removed from the partition to the LSM(s) that will remain in the partition

- When any LSM(s) is removed from an ACS, `moving.sh` can move cartridges to the LSMs that will remain in the ACS.

  For example, if SL8500s are removed from a library complex (ACS), `moving.sh` moves cartridges from the SL8500s that are being removed to the LSMs that will remain in the library. This also applies when 9310 LSMs are removed from an ACS of 9310s.

- When a storage expansion module(s) is removed from an SL8500, cartridges can be moved from the panels being removed to the panels that will remain in the library

- To optimize library performance you may need to move inactive cartridges to an LSM with few or no drives that are used to archive cartridges. This frees up space in LSMs with drives for new, active cartridges.

Creating the Volume List File

Before you begin, you must create a file that contains the list of volumes to be moved from an LSM. You can use either the *volrpt* (custom volume report) or the `display volume` command.

- Using `volrpt` to create *vol_list_file*

  ```
  volrpt -d -f custom_volrpt_input -l  lsm_id > vol_list_file
  ```

  Where the *custom_volrpt_input* file is:

  > VOLUME_ID 6

  Sample output

  ```
  $ volrpt -d -f my_custom -l 0,2 > my_file_list
  $ cat my_file_list
  ABC744
  ABC748
  ABC756
  ACS151
  EN0823
  O00373
  ```

- Using the `display volume` command to create the *vol_list_file*

a. Display the list of volumes.

Example:

```
display volume * -home acs,lsm,panel,*,* -f vol_id
```

This example selects all volumes on the panel identified by the -home parameter. Row and column are wild-carded. Only the *vol_id* is output.

Sample output

```
ACSSA> display volume * -home 0,3,5,*,* -f vol_id
2007-02-12 15:31:45              Display Volume
Vol_id
PG0350
PG0353
PG0356
PG0358
PQ0616
```

b. Create and name your *vol_list_file*.

c. Cut and paste the list of volumes (created from the display command) into this file.

d. Edit the output.

The *vol_list_file* cannot contain any blank lines and leading spaces. Use to following vi command to eliminate them.

:%s/^[ ]*//g

If you do not do this, you get an error message as shown in the following example.

```
$ moving.sh -f my_file_list -t 0,2

  Error in file my_file_list.
  Invalid entry
 ABC748
 ABC756
ACS151
EN0823
```

This error message was generated because there was an extra space before volumes ABC748 and 756.

Procedures for Moving a Group of Cartridges

The following procedures describe how to:

• move cartridges before removing an LSM from an ACS

• move cartridges before changing or removing panels

Moving Cartridges Before Removing an LSM from an ACS

After a library is re-configured or re-partitioned and if an LSM is removed from an ACS, all cartridges in the LSM become inaccessible. So, before the LSM is removed, all its cartridges should be moved to LSM(s) that will remain in the ACS. Use the following procedure:

- When a rail (LSM) is removed from a partition in a partitioned SL8500.

- When an LSM(s) is removed from an ACS. The ACS can include 9310s or an SL8500 library.

1. Plan your new configuration.

    - Organize the cartridges and drives for performance.

    - Empty an LSM shortly before you change the library configuration.

    - Determine how many cartridges you have in the LSMs that you are emptying, and how many free cells in the LSMs to which you are moving cartridges.

      Use `free_cells.sh -a` to find out the number of cartridges in these LSMs (allocated cells) and free cells.

2. Schedule the move and reconfiguration.

    - Schedule the move to minimize the impact on your system.

      Moving the cartridges takes time, and reconfiguring a library or repartitioning an SL8500 is disruptive.

    - Make sure there are enough free cells in the target LSM(s) for the cartridges being moved. If you have to, eject cartridges to free up space.

3. Vary all of the drives in the LSM being removed offline.

    This prevents:

    - Contention for robots in the LSM.

    - Mounts to this LSM.

      Otherwise, cartridges mounted to this LSM can float to new home cells in the LSM, filling up the LSM that you are trying to empty.

4. Vary the LSM being emptied to diagnostic mode to restrict access to only the cmd_proc using:

    `vary lsm *lsm_id* diagnostic`

    Example: `vary lsm 0,1 diagnostic`

5. Run a custom volrpt to output all of the cartridges in the LSM being emptied to a file, using the following command:

```
volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list
```

Where the `custom-volrpt_input` is:

VOLUME_ID 6

Example:volrpt -f volrpt_input -l 0,1 > move_vols_list

Refer to for more information.

6. Move the cartridges out of the LSM being emptied, using the following:

```
moving.sh -f move_vols_list -t dest_lsm_id(s)
```

7. Check that the LSM is empty using `volrpt` since cartridges may have been entered into the LSM or may have "floated" into it.

```
volrpt -l from_lsm_id
```

If it is not empty run the custom volrpt again to select the volumes that are now in the LSM. Then, run `moving.sh` again (steps 3 and 4).

Note: **Do not** run `moving.sh` again with the original list of volumes.

8. Vary the LSM being emptied offline to prevent volumes from being moved to it.

```
vary lsm lsm_id offline
```

9. Remove the LSM from the partition and/or ACS.

10. Reconfigure the ACS, using either `config acs acs_id` or `acsss_config`.

Moving Cartridges Before Changing or Removing Panels

You need to move the cartridges before changing a cell panel to a drive panel in a 9310 or removing a storage expansion module in an SL8500.

Steps 1 - 4 as procedures as.

Step 5: Select the cartridges in the panel being emptied, and output them to a file.

a. Run a custom volrpt to output all of the cartridges in the LSM being emptied to a file. Include the panel number (in the home cell ID).

```
volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list_1
```

Where the `custom-volrpt_input` is:

VOLUME_ID 6
CELL_ID   14

Select the volumes in the panel(s) being emptied and output these vol_ids to your `move_vols_list_2`.

b. Select the cartridges in a panel being emptied using the `display volume` command.

display volume * -home acs,lsm,panel,*,* -f volume > *move_vols_list_2*

This selects all volumes on the panel identified by the *-home* parameter. The row and column are wild-carded. Only the *vol_id* is output, and the output is written to the file.

Edit the output, removing any leading spaces and the trailing blank line.

Note: If the destination or "to" LSM is the same as the source or "from" LSM and more than one panel is being emptied, some of the volumes will be moved back to the panels being emptied. You will have to select the volumes off the panels and move them repeatedly to clear out the panels.

Steps 6 - 9 same procedures as "Moving Cartridges Before Removing an LSM from an ACS" on page 161.

Step 10. Reconfigure the LSM, using either `config lsm` *lsm_id* or `acsss_config`.

Examples

- Moving cartridges from LSM 0,4 to LSM 0,0 and 0,1

To move cartridges from LSM 0,4 to LSM 0,0 and 0,1, you must first create a file containing the list of cartridges in LSM 0,4 using *volrpt*, and then run the `moving.sh` utility as below**:**

Sample output:

```
$ moving.sh -f vol_list.txt -t 0,0 0,1

Number of free cells in LSM 0,0 : 308
Number of free cells in LSM 0,1 : 362
----------------------------------------
Total number of free cells : 670
Total number of volumes to move : 7

Cartridge CAB001 moved to 0,0,3,0,0
Cartridge CAB002 moved to 0,0,4,0,0
Cartridge CAB003 moved to 0,0,5,0,0
Cartridge CAB004 moved to 0,0,6,0,0
Cartridge CAB005 moved to 0,0,7,0,0
Cartridge CAB006 moved to 0,0,8,0,0
Cartridge CAB007 moved to 0,0,9,0,0


Summary

=======

Number of free cells remaining in LSM 0,0 : 301
Number of free cells remaining in LSM 0,1 : 362
```

```
         -------------------------------------------------
         Total number of free cells remaining : 663
         Number of cartridges moved : 7
         Number of cartridges not moved : 0
```

- Moving cartridges from LSMs 0,4 0,5 0,6 and 0,7 to LSMs 0,0 0,1 0,2 and 0,3,

  To optimize performance by moving each LSM to the adjacent LSM:

  1. Prepare files containing the list of cartridges in LSM 0,4 0,5 0,6 and 0,7 using *volrpt.*

  2. Run four `moving.sh` utilities at the same time but in separate Unix command terminals.

     There is no contention between the separate instances of `moving.sh` because the source and destination LSMs and the pass-thru ports used are all different:

  Sample output

  ```
  moving.sh -f vol_list_0-4.txt -t 0,0
  moving.sh -f vol_list_0-5.txt -t 0,1
  moving.sh -f vol_list_0-6.txt -t 0,2
  moving.sh -f vol_list_0-7.txt -t 0,3
  ```

Managing Cartridges for Performance

The `moving.sh` utility can be used to move inactive cartridges to archival LSMs. An archival LSM is an LSM with few or no drives that is used to store cartridges that have a low probability of being mounted. The top rail in an SL8500 is a good choice for an archival LSM because it does not have direct access to the CAP.

Inactive cartridges that do not need to be in a library can be ejected, while inactive cartridges that still need to be available for automated mounts should be moved to archival LSMs.

To move inactive cartridges to an archival LSM:

1. Identify the inactive cartridges. For example to select cartridges that have not been accessed in the last three months:

2. Run a custom volrpt to output all of the cartridges in the LSM being examined, and output the results to a file.

   ```
   volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list_1
   ```

   Where the *custom-volrpt_input* is:

   VOLUME_ID 6

ACCESS_DATE 15

3. Select the cartridges where the access_date is earlier than three months ago and output these vol_ids to a file with the list of volumes to be moved.

4. Move the inactive cartridges to the archival LSM.

    `moving.sh` *move_vols_list_2 archival_lsm_id*

See also

-
-

■ probeFibre.sh

This utility displays all attached libraries connected behind a Sun-branded HBA, or behind a contemporary Emulex or QLogic HBA. In the case of native Emulex or QLogic cards, probeFibre.sh relies on the Emulex 'hbacmd' or the QLogic 'scli' utility to collect the vital product data. The `probeFibre.sh` utility displays the model number, revision level and *target:lun i.d.* of each fibre-attached library. Using the *-v* option, you can display the model number of the attached HBA. This utility requires *root* access.

The `probeFibre.sh` utility can be run even before the mchanger devices are created for each library.

Format

`probeFibre.sh`

■ rc.acsss

The `rc.acsss` utility brings ACSLS up and automatically starts the database. There is no need to manually start the database.

Format

`rc.acsss [idle]`

Options

`idle` brings ACSLS up in the idle state.

Usage

Use the `rc.acsss` utility to start ACSLS. When you start ACSLS, it goes through recovery mode to online state. ACSLS initialization includes checking its database and "recovering" status of the library cartridges and library hardware.

See Also

Note: If you are not logged in as `acsss` and try to run `rc.acsss`, the system prompts you for the `acsss` password.

You can run only one copy of ACSLS on the server. `rc.acsss` will not restart ACSLS if it is already running.

Note: `rc.acsss` automatically starts the database when it starts ACSLS.

■ rdb.acsss

The `rdb.acsss` utility restores the ACSLS database and the ACSLS Control files using a backup created by either the automatic backup function or the `bdb.acsss` utility. The ACSLS Control files are located in `$ACS_HOME/data`, and define several different environmental variables for ACSLS. They specify Access Control settings, scratch preferences, Extended Store LSMs, custom `volrpt` settings, and volume attributes (for `watch_vols` utility), etc.

Note: You need to rewind or position the tape device before you restore the ACSLS database and control files from tape. To do this, rewind or position the tape to the exact location where the backup files resided before running `rdb.acsss`.

- For Solaris use one of the following:

  - `mt -f /dev/rmt/0mn rewind`

  - `mt -f /dev/rmt/0mn nbsf 1`

- For AIX use one of the following:

  - `mt -f /dev/rmt1.1 rewind`

  - `mt -f /dev/rmt1.1 bsf 1`

Format

```
rdb.acsss
```

Menu Options

When you run rdb.acsss, a menu displays four options:

Figure 15. rdb.acsss Menu Options

```
Please enter the number followed by Return for your choice from

the following menu.


Press? followed by the Return key for help.


    1: Restore from a current local disk backup

    2: Restore from a previous tape or file backup

    3: Restore database only (do not include ACSLS control files)

    4: Restore only ACSLS non-database control files

    E: Exit
```

1. Restore from a current local disk backup

Explanation: The database is restored to the backup. ACSLS Control files are restored from the backup only. The backups are saved in the default backup directory ($ACSDB_BKUP). The database is restored to any database backup listed and selected. Usually there are 8 different dates listed, but this varies depending on the database retention period set in acsss_config.

Usage: Use this option to restore a corrupted database. With this option, all backups are displayed and you can restore to any displayed database backup.

Example:

```
Menu choice: 1

rcvr_previous.sh 2642:  ACSLS database recovery started.

You have taken backups on the following days. Please enter the
corresponding date and time to the backup that you wish to recover
from. ACSLS database and control files will be restored to that
time.

                    2007-02-02 04:38:48

                    2007-02-03 00:00:01

                    2007-02-04 00:00:01

                    2007-02-05 00:00:01

                    2007-02-05 11:49:06

Please enter the recovery date and time (YYYY-MM-DD HH:MM:SS):
```

```
HINT: You may copy and paste to enter the date and time.
```

You need to enter the desired date and time from the relevant backup and the database is restored to that point.

2. `Restore from previous tape or file backup`

   Explanation: Select this option to recover a database that was copied to a different file system (such as NFS) or to a backup device (such as tape). ACSLS Control files are restored.

   Usage: Use for a catastrophic event such as hardware failure when the database needs to be restored to either the server or even an entirely different server. The platform (OS version/update and ACSLS release/PUT level) must be the same.

   Option 2:

   ```
   Menu choice: 2.

   rcvr_manual.sh 2635: ACSLS recovery started

   To recover the ACSLS environment either:

   - Mount a ACSLS backup tape in a tape device and

     specify this tape device with '-f tape_device', or

   - Specify a file name containing a ACSLS backup with '-f
     backup_file'.

   The ACSLS database will be recovered from the file specified.

   Please enter -f [ backup_file | tape_device ]:
   ```

   Example 1: Specifying a file with '-f *backup_file*'

   ```
   Please mount tape (if used) and enter backup source: -f /export/
   backup/my_backup.bak
   ```

   This would restore a backup called `my_backup.bak`. Both database and ACSLS Control files would be restored and ACSLS would be put back in the state that it was when the backup was run.

   Example 2: Restoring a backup created on a tape device

   Restoring a backup created on a tape device uses the same option but works a little differently. When a backup is created to a tape device, the tar archive is created on the tape but it doesn't have a name. When restoring a backup from a tape, only the tape device is given.

   Hint: You should use a **no rewind** tape device.

   ```
   Please mount tape (if used) and enter backup source: -f /dev/rmt/
   0mn
   ```

   This goes to the device "/dev/rmt/0mn" and verifies there is a valid database backup. It it does exist and is valid, it is restored.

Procedure to Rewind a Tape:

The tape needs to be REWOUND or POSITIONED at the correct location where the backup files reside before rdb.acsss' is attempted.

Note: After running "tar tvbf" command, the tape is advanced to the next block. Make sure you have rewound/repositioned the tape if you intend to run `rdb.acsss` after running "tar tvbf" command.

a. The tape can be rewound/positioned using:

mt -f /dev/rmt/0mn rewind" or "mt -f /dev/rmt/0mn nbsf 1 ---> SOLARIS
mt -f /dev/rmt1.1 rewind" or "mt -f /dev/rmt1.1 bsf 1 ---> AIX

b. To verify, the contents of tape after `bdb.acsss`, use the following commands:

tar tvbf 2048 /dev/rmt/0mn ---> SOLARIS
tar tvbf 20 /dev/rmt1.1 ---> AIX

3. `Restore database only (do not include ACSLS control files)`

Explanation: The option provides to ability to restore data only. In some environments you may need to restore the ACSLS database including its data, but you do not need to restore the ACSLS non-database control files.

Option 3:

```
Menu choice: 3

To recover the ACSLS database data only either:

- Mount an ACSLS backup tape in a tape device and

  specify this tape device with '-f tape_device', or

- Specify a file name containing an ACSLS backup with '-f
backup_file'.

The ACSLS database data will be recovered from the file specified.

   ****This option does NOT include the ACSLS control files****

Please enter -f [ backup_file | tape_device ]:
```

4. `Restore only ACSLS non-database control files`

Explanation: Restores only the ACSLS control files. Prior to restoring any file located in the $ACS_HOME/data/internal directory, backups will be made of the existing files, appending the end with a ".bak" extension.

```
$ACS_HOME/data/internal/dynamic_variables/dv_config.dat.bak
$ACS_HOME/data/internal/dynamic_variables/dv_trace.dat
$ACS_HOME/data/internal/release.vars.bak
```

This is not the case for files located in $ACS_HOME/data/external. No backups are performed of the ACSLS control files prior to recovery.

Option 4:

```
Menu choice: 4

To recover the ACSLS non-database control files either:

- Mount an ACSLS backup tape in a tape device and

  specify this tape device with '-f tape_device', or

- Specify a file name containing an ACSLS backup with '-f
backup_file'.

ACSLS non-database control files will be recovered from the file
specified.

Please enter -f [ backup_file | tape_device ]:
```

Example:

```
Please enter -f [ backup_file | tape_device ]: -f /export/home/
my_file.bak
```

- `-f /export/home/my_file.bak` recovers the ACSLS control files from the specified file

- `-f /dev/rmt/0mn` recovers ACSLS control files from the specified tape device

5. `Exit`

    When you exit the `rdb.acsss` utility, a backup in initiated to the default directory, `$ACSDB_BKUP`.

See Also

Recovery procedures for:

- "Restoring the Database to the Most Recent Backup" on page 117
- "Recovering from a Failed Server" on page 117
- "bdb.acsss" on page 131

■ showDevs.sh

The `showDevs.sh` utility displays the critical device attributes associated with each mchanger instance in the `/dev` directory. Critical attributes include the library model number, cell capacity, number of attached drives, and Target-LUN address. The utility also displays the model number of the host bus adapter that is associated with each mchanger device. Additional attributes can be displayed using the following options.

Format

```
showDevs.sh [-s][-w][-q][-sq][-wq]
```

Usage

The utility can be run as `root` or as `acsss`. Each user ID provides a different set of capabilities. For example:

- `acsss`

  user `acsss` can list the serial number and the number of cells and drives associated with each library

- `root`

  `root` can extract serial number and the number of cells and drives only for libraries that are not online to ACSLS.

  However, `root` is able to reveal the Wide Node Name and World Wide Port Name of each fibre-attached library.

Options

This utility can be run with several options.

no argument
: reveals each HBA and the model, revision level, target and LUN of each attached library.

`-s`
: Optional. This option displays the serial number of each library along with the number of cells and the number of attached drives associated with each library.

`-w`
: Optional. This option displays the World Wide Names (WWNN and WWPN) for each HBA and for each attached target library.

  The "-w" option is not available for user `acsss`.

`-q`
: Optional. This option displays only essential mchanger data, while excluding information about the HBAs.

`-sq`
: Optional.

`-wq`
: Optional.

Examples

```
# whoami
root

# pwd
/export/home/ACSSS/diag/bin
```

```
#   ./showDevs.sh
   QLogic QLA2300   QLGC,qla@1
         /dev/mchanger0: STK L700 V-1012 Target-3 LUN-0
         /dev/mchanger1: STK L700 V-1012 Target-3 LUN-1
         /dev/mchanger2: STK VLIBRARY V-1012 Target-4 LUN-1
         /dev/mchanger3: STK VLIBRARY V-1012 Target-4 LUN-2

    LSILogic 53c1030  LSILogic,scsi@1
         /dev/mchanger4: STK SL500 Target-0 LUN-0

    Emulex LP10000  fibre-channel@2
         /dev/mchanger5: STK L180 Target-0 LUN-0


# ./showDevs.sh -q
         /dev/mchanger0: STK V700 216-cells 10-drives
         /dev/mchanger1: STK V700 384-cells 10-drives
         /dev/mchanger2: STK VLIBRARY 1000-cells 50-drives
         /dev/mchanger3: STK VLIBRARY 10000-cells 461-drives
         /dev/mchanger4: STK SL500 Target-0 LUN-0
         /dev/mchanger5: STK L180 Target-0 LUN-0


# ./showDevs.sh -wq
         Target-3 WWNN-50-01-04-F0-00-7E-B3-39 WWPN-50-01-04-F0-00-7E-B3-3A
         /dev/mchanger0: STK L700 V-1012 Target-3 LUN-0
         /dev/mchanger1: STK L700 V-1012 Target-3 LUN-1

         Target-4 WWNN-50-01-04-F0-00-7E-B3-45 WWPN-50-01-04-F0-00-7E-B3-46
         /dev/mchanger2: STK VLIBRARY V-1012 Target-4 LUN-1
         /dev/mchanger3: STK VLIBRARY V-1012 Target-4 LUN-2

         /dev/mchanger4: STK SL500 Target-0 LUN-0

         Target-0 WWNN-50:01:04:F0:00:41:18:46 WWPN-50:01:04:F0:00:41:19:32
         /dev/mchanger5: STK L180 V-0310 Target-0 LUN-0


#   ./showDevs.sh -w
   QLogic QLA2300   QLGC,qla@1
         HBA:  WWNN 20-00-00-E0-8B-07-AA-3A  WWPN 21-00-00-E0-8B-07-AA-3A

         Target-3 WWNN-50-01-04-F0-00-7E-B3-39 WWPN-50-01-04-F0-00-7E-B3-3A
         /dev/mchanger0: STK L700 V-1012 Target-3 LUN-0
         /dev/mchanger1: STK L700 V-1012 Target-3 LUN-1

         Target-4 WWNN-50-01-04-F0-00-7E-B3-45 WWPN-50-01-04-F0-00-7E-B3-46
         /dev/mchanger2: STK VLIBRARY V-1012 Target-4 LUN-1
         /dev/mchanger3: STK VLIBRARY V-1012 Target-4 LUN-2

    LSILogic 53c1030  LSILogic,scsi@1

         /dev/mchanger4: STK SL500 Target-0 LUN-0

   Emulex LP10000  fibre-channel@2
         HBA: WWNN 20:00:00:00:c9:46:57:b3  WWPN 10:00:00:00:c9:46:57:b3

         Target-0 WWNN-50:01:04:F0:00:41:18:46 WWPN-50:01:04:F0:00:41:19:32
         /dev/mchanger5: STK L180 V-0310 Target-0 LUN-0


# su - acsss
$ whoami
acsss
```

```
$ showDevs.sh -sq
        /dev/mchanger0: STK V700 S/N-00043000022 216-cells 10-drives
        /dev/mchanger1: VL700 S/N-00043000040 10-drives in use by ACSLS.
        /dev/mchanger2: STK VLIBRARY S/N-00043000037 1000-cells 50-drives
        /dev/mchanger3: STK VLIBRARY S/N-00043000045 10000-cells 461-drives
        /dev/mchanger4: SL500 S/N-52200000032 6-drives in use by ACSLS.
        /dev/mchanger5: L180 S/N-MPC01000124 4-drives in use by ACSLS.


$ showDevs.sh -s
  QLogic QLA2300  QLGC,qla@1
        /dev/mchanger0: STK V700 S/N-00043000022 216-cells 10-drives
        /dev/mchanger1: VL700 S/N-00043000040 10-drives in use by ACSLS.
        /dev/mchanger2: STK VLIBRARY S/N-00043000037 1000-cells 50-drives
        /dev/mchanger3: STK VLIBRARY S/N-00043000045 10000-cells 461-drives


  LSILogic 53c1030  LSILogic,scsi@1
        /dev/mchanger4: SL500 S/N-52200000032 6-drives in use by ACSLS.

  Emulex LP10000  fibre-channel@2
        /dev/mchanger5: L180 S/N-MPC01000124 4-drives in use by ACSLS.
```

■ stats_report

The `stats_report` utility generates library volume statistics reports. To run this utility, you must be logged in as `acsss`.

Format

```
stats_report [vol_statsX.log ...]
```

Where:

`vol_statsX.log` -

1. .Using this optional argument, you can specify one or more archived volume statistics log file names.

    (The archived files have the format vol_statsX.log (where 0 <= X <= 8).)

    Using one archived file as input:

        `$stats_report  vol_stats0.log`

    The time centric and drive centric reports are generated with the name of the user input file pre-pended to (before) the report file name.

    For example, if you specify `vol_stats0.log` then the reports will be generated in the `$ACS_HOME/log` directory as

    `vol_stats0_drive_centric.txt` and
    `vol_stats0_time_centric.txt`

2. To generate a report for all the arcguved volume stats files at once, follow the procedure below:

    a. Generate the full log from individual files

        ```
        $cd $ACS_HOME/log
        $cat vol_stats8.log .... vol_stats0.log
           acsss_stats.log  >  vol_statsXXXX.log
        ```

        where `vol_statsXXXX.log`

        (The string `vol_stats` is necessary, but *XXXX* can be anything like FULL,etc) is the concatenated file of all vol_statsX.log(where 0 <= X <= 8) and acsss_stats.log in reverse order.

    b. Run `stats_report.`

        ```
        $stats_report  vol_statsXXXX.log
        ```

        Reports are generated as `vol_statsXXXX_drive_centric.txt` and `vol_statsXXXX_time_centric.txt.`

        If no filename is given as an argument, then the time centric and drive centric reports are generated from `$ACS_HOME/log/acsss_stats.log.`

Usage

- The stats_report uses the current acsss_stats.log to prepare two reports of volume statistics. You must enable library volume statistics gathering by setting the variable `LIB_VOL_STATS` to on. This can be done through the `acsss_config` process or through the command line command `dv_config -p LIB_VOL_STATS`. ACSLS then automatically rolls and maintains 9 `acsss_stats.log` files when the log reaches the default size of 500 KB.

- The size of the log files and the number of files to retain is controlled through the variables LIB_STATS_FILE_NUM and VOL_STATS_FILE_SIZE. These variables are set using the same method as LIB_VOL_STATS discussed above.

- The two types of reports are:

    - drive_centric.txt

    This report contains an ordered list of drives. Each drive record contains all cartridges mounted to the drive, the requestor, the time of the request and the duration of the mount.

    - time_centric.txt

    This report contains the usage of drive resources listed on an hourly time scale. Each record in a time period includes the requestor, the

specific drive, the number of mounts during that period for that drive, and the duration of drive usage during the hour. Note: if the drive usage exceeds 60 minuets for a time period it is an indication that the mount spanned two time periods and the mount will not be listed in the second time period.The first report created by `stats_report` is a drive view.

Notes on processing details:

- If there is a DISMOUNT record in the log but there is no corresponding MOUNT record, the reason could be that:

  - The log was rolled over, or

  - The operation was logged due to some unknown logging problem.

  In this case the record is omitted from the report generated.

- If there is a MOUNT record in the log but there is no corresponding DISMOUNT record, the reason could be that:

  - The DISMOUNT has not yet happened, or

  - The operation was not logged due to some unknown logging problem.

  In this case the mount duration is set to -1, which is an indication of cases mentioned above. These records are omitted from calculating the total mount duration in the time centric report.

- In the cases of going from day light savings time to standard time, scenarios where the calculated mount duration is negative can arise. To suppress them, the absolute value of the mount duration is taken.

■ volrpt

The `volrpt` utility creates a volume report.

Format

```
volrpt [-s vol|loc|use] [-d] [-f filename] [-z] [-a|-l|-v
identifier_list] [-i]
```

Options

`-s`

specifies the sort order. If you do not specify this option, the default is to sort by volume ID. If you specify this option, you must specify one of the following values:

`vol`
  sort by volume ID.

`loc`

      sort by volume home location.

  use

      sort by volume use (number of mounts).

-d

specifies that the output contains no page breaks or header information. The output can be used as input to other programs such as `pr`.

-f *filename*
filename specifies a custom volrpt template.

-z

zero fills identifier fields.

-a

restricts the report to the specified ACS. You can specify multiple ACSs (use blanks to separate the *acs_ids*).

-l

restricts the report to the specified LSM. You can specify multiple LSMs (use blanks to separate the lsm_ids).

-v

restricts the report to the specified volumes (or volume ranges). You can specify a single vol_id, a list of vol_ids separated by blanks, or a volume range indicated by vol_id-vol_id.

-identifier_list
described by the -v, -a, and -l options. This is a list of ACSs, LSMs, and volumes (or volume ranges).

-i

reports all volumes, including absent and ejected cartridges.

If this option is not specified, absent and ejected cartridges are not reported.

Usage

Use the `volrpt` utility to create a report of library cartridges, including their physical location, history, attributes, and use. You can also use volrpt to verify the database after you restore it. You can use the `-a`, `-l`, or `-v` options to specify the ACSs, LSMs, or cartridges for the report. If you do not specify any of these options, `volrpt` reports on only ACS 0.

Figure 16 shows an example of a standard volume report, which contains fields for volume id, location, label type, media type, and history of usage.

Figure 16. Standard Volume Report

```
                          VOLUME REPORT UTILITY


2002-06-30 14:01:21


TOTAL VOLUMES: 400 SEQUENCE: sort by volume identifier


Volume Home     Label    Volume    Times   |---Entered---||--Last Used--|
Label  Location Attr     Type/Media Mounted Date      Time      Date     Time


CLN000 0,0,1,0,3 Ext     C/STK1U   1        08/22/01 09:30      10/04/01 14:26
RB0000 0,1,2,1,10Ext.    D/STK1R   3        10/01/01 08:16      10/01/01 08:18
RB1400 0,0,10,1,3Ext.    S/STK1R   243      10/01/01 09:30      10/06/01 11:04
RB1401 0,0,10,3,5Virt.   D/STK1R   12       10/01/01 03:29      10/05/01 23:11
  "      "      "         "     "           "        "
  "      "      "         "     "           "        "
  "      "      "         "     "           "        "
TB1440 0,1,3,1,9 Ext.    D/STK2P   43       08/12/01 09:11       09/28/0117:52
  "      "      "         "     "           "        "
  "      "      "         "     "           "        "
  "      "      "         "     "           "        "
```

In the `Volume Type/Media` column, C denotes cleaning cartridges, D denotes data cartridges, and S denotes scratch cartridges.

Use the `-f` *filename* option to create a customized report; see "Creating a Custom Volume Report" on page 121, for more information.

`$ACS_HOME/data/external/volrpt/owner_id.volrpt` is a sample input file that you can run or use as a template to create customized volume reports. You can also save your customized volume reports in the `$ACS_HOME/data/external/volrpt` directory.

You can redirect the volume report to a file with standard UNIX redirection:

`volrpt >` *file*

Examples

By default, volrpt reports only the first ACS in the list. To report the cartridges in both ACS 0 and ACS 1, enter:

`volrpt -a 0 1`

To report the cartridges in LSMs 0,1 and 2,1 sorted by home cell location, enter:

```
volrpt -s loc -l 0,1 2,1
```

Notes

- `volrpt` displays the specified volume report if it completes successfully. `volrpt` prints a message to **stderr** and exits if you specify the `-f` option and volrpt cannot find the specified file or you specify more than one input file. For field errors within the input file, `volrpt` prints a message to `stderr` and ignores the line in error but does not exit.

- If cartridges are not found in the volume ID list or range or library component you specified, `volrpt` returns a `no volumes found` message.

- When a parameter is not specified, it uses the default of ACS 0.

- If a library component(s) is specified through the -a, -l, or -v option, but no volumes are found, messages such as the following are displayed:

  - **-a** option (ACS)

    Messages:

    when a single acs_id is provided and no volumes are present, the following error displays: `No Volumes found for ACS: (<acsid>)`

    Example:

    ```
    $ volrpt -a 2
    No Volumes found for ACS: (2)
    ```

    when multiple acs_ids are provided and none of them have any volumes, the following error displays: `No Volumes found for ACS: (<acsid1>)(<acsid2>)`

    Example:

    ```
    $ volrpt -a 0 1
    No Volumes found for LSM: (0) (1)
    ```

  - **-l** option (LSM)

    Messages:

    when a single lsm_id is provided and no volumes are present, the following error displays: `No Volumes found for LSM: (<lsmid>)`

Example:

```
$ volrpt -l 1,1
No Volumes found for LSM: (1,1)
```

when multiple lsm_ids are provided and none of them have any volumes, the following error displays: `No Volumes found for LSM: (<lsmid1>)(<lsmid2>)`

Example:

```
$ volrpt -l 1,1 1,2
No Volumes found for LSM: (1,1) (1,2)
```

- **-v** option (VOLUME)

Messages:

when a single `volid` is provided and no volumes are present, the following error displays: `Volume(s) not: (<volid>)`

Example:

```
$ volrpt -v BBB112
No Volumes found: (BBB112)
```

when multiple volids are provided and none of them have any volumes, the following error displays: `Volume(s) not found: (<volid1>)(<volid2>)`

Example:

```
$ volrpt -v BBB112 BBB114
No Volumes found: (BBB112) (BBB114)
```

The **-v** option can also be used for volume range, and produces similar messages when no volumes are present.

when a single volume range is provided and no volumes are present, the following error displays: `Volume(s) not: (<volrange>)`

Example:

```
$ volrpt -v BBB112-BBB116
No Volumes found: (BBB112-BBB116)
```

when multiple volume range is provided and no volumes are present, the following error displays: `Volume(s) not: (<volrange1>) (<volrange2>)`

Example:

```
$ volrpt -v BBB112-BBB116 BBB220-BBB224
No Volumes found: (BBB112-BBB116) (BBB220-BBB224)
```

- When an ACS or LSM has not been configured

  When volrpt is used with an acs_id or lsm_id that does not exist, it displays a message according to the identifier.

  - **-a** (ACS)

    ```
    ACS identifier (<acsid>) not configured
    ```

  - **-l** (LSM)

    ```
    LSM identifier (<lsmid>) not configured
    ```

See Also

-

■ watch_vols

This utility applies a pre-defined policy to newly entered or newly discovered volumes in an ACSLS library. That policy is defined in the file:

```
$ACS_HOME/data/external/vol_attr.dat
```

This file contains a list of user-defined volume IDs or volume ranges and a user-specified policy with each volume entry. For each volume or volume range listed in that file, you can define volume ownership, pool association and/or preferred LSM location. Specific instructions for defining policies are explained in detail in vol_attr.dat.

The watch_vols utility uses the acsss_stats.log to discover the existence of newly entered volumes or volumes discovered during an audit. To enable this capability, you must enable volume statistics with acsss_config. With volume statistics enabled, watch_vols monitors the tail of the acsss_stats.log, looking for matching volumes with the entries defined in vol_attr.dat. Wherever a match is found, the defined policy for that volume is automatically applied.

Note: The volume IDs must follow these rules:

- One cartridge-id per line.

- The vol_ids must be valid ACSLS volume IDs.

- If the vol_ids include trailing or leading spaces, they must be represented as underscores (_).

Format

```
watch_vols [start|stop]
```

Usage

You can check the running status of the utility by invoking `watch_vols` with no parameter. If you are unsure of the status of `watch_vols` (running or stopped), the command `watch_vols` with no argument will display the current status.

There are two options for `watch_vols`, *start* and *stop*.

- `watch_vols start`

  When the start parameter is invoked, `watch_vols` reviews the policy defined in `vol_attr.dat`. If there are errors in format or syntax, `watch_vols` displays the error and prompts you to make the necessary correction to `vol_attr.dat`. Once the defined policy is accepted by `watch_vols`, the utility invokes a daemon to run in the background. The daemon continues to run as long as ACSLS is running. It starts automatically whenever ACSLS is restarted.

  The policy table in `vol_attr.dat` can be updated at any time. You need not stop `watch_vols` in order to update the policy. Just run `watch_vols start` to commit the updates to the running program.

- `watch_vols stop`

A log of all `watch_vols` activities is maintained in the log file

$ACS_HOME/log/watch_vols_event.log

Each change of volume ownership, pool_id, or LSM home location is logged in this file.

Examples

- You want to move specific volumes to a target LSM during an initial audit

  1. Audit the target LSM with `watch_vols` **disabled**.

  2. Once the target LSM has been audited, start `watch_vols.`

  3. Perform a general audit`.`

     Audit then migrates the specified volumes to the destination LSM.

316120101

Command References

In this chapter, you will learn how to use the ACSLS commands, including general command syntax and reference information.

"set commands" on page 259
> Sets various attributes of different library components.

"show" on page 270
> Displays your lock ID or user ID.

"start" on page 271
> Starts ACSLS request processing.

"switch lmu" on page 272
> Manually switches ACS management from the ACS's master LMU to the standby LMU.

"unlock" on page 275
> Removes active locks on cartridges or drives.

"vary" on page 277
> Changes the state of an ACS, LSM, CAP, transport, or port.

"venter" on page 284
> Enters one or more cartridges with missing or unreadable labels into the ACS.

> **Note:** The L5500, SL500, and SL8500 libraries do not support unlabeled cartridges (venters).

■ General Command Syntax

This section describes the general syntax of the ACSLS commands. The following sections fully describe each command, including its syntax.

ACSLS commands use the following general syntax:

```
command type identifier state options
```

Where:

type identifier
> is the ACS component and its identifier; see "Component Types and Identifiers" on page 185 for more information.

state
> is a device state for the <u>va</u>ry command only.

options
> are command options; see the description of the command you want to run.

Also note the following syntax rules:

• You must enter commands in the order shown above (command name, followed by the component and its identifier, followed by the state and any options).

- This chapter shows commands in lowercase, but you can enter commands in any combination of lowercase and uppercase letters.

- Underlines show minimum abbreviations for commands and keywords. For example, `query server` and `q ser` are both valid forms of the `query server` command.

- Ellipses (...) indicate that you can repeat an identifier.

- Brackets [ ] enclose optional options.

■ Component Types and Identifiers

Table 16 describes the valid ranges of values for each of the ACS component identifiers. You can specify a maximum of 42 identifiers for each type. See specific command descriptions for the valid component types for each command.

Note: The identifiers specified in Table 16 represent valid ranges of values supported by the software. Your LSM type and library configuration determine what identifier values are valid for your particular site.

Table 16. ACSLS Component Types and Identifiers

Component	Type	Identifier	Valid Values
entire library	`server`	none	none
ACS	`acs`	*acs_id*	acs(0‑31), lsm(0‑99)
LSM	`lsm`	*lsm_id*	acs(0‑31), lsm(0‑99)
LSM panel	`panel`	*panel_id*	acs(0‑31), lsm(0‑99), panel(0‑50),
LSM subpanel *	`subpanel`	*subpanel_id*	acs(0‑31), lsm(0‑99), panel(0‑50), startrow(0‑51), startcolumn(0‑23), endrow(0‑51), endcolumn(0‑23)

Table 16. ACSLS Component Types and Identifiers

Component	Type	Identifier	Valid Values
LSM storage cell	subpanel	*cell_id*	acs(0-31), lsm(0-99), panel(0-50), row(0-51), column(0-23)
* The ending row (and column) must be greater than or equal to the beginning row (and column).Only cells within the beginning and ending matrix are audited: the matrix starts with the beginning row and beginning column and extends to the ending row and ending column.			
CAP	cap	*cap_id*	acs(0-31), lsm(0-99), cap(0-11) An asterisk (*) in the *cap_id* does the following: acs,lsm,* - causes ACSLS to select the highest priority available CAP in the LSM. acs,* - causes ACSLS to select the highest priority available CAP in the ACS. * - for an enter request causes ACSLS to select the CAP in the LSM with the most free cells. * - for an eject request causes ACSLS to select the highest priority CAP in each ACS with a cartridge designated for ejection.
transport	drive	*drive_id*	acs(0-31), lsm(0-99), panel(0-50), drive(0-31)
drive type	drive	*drive_type*	10-character drive type identifier; can be any combination of numbers (0-9) or letters (A-Z). Spaces are not allowed.
port	port	*port_id*	acs(0-31), port(0-3)

Table 16. ACSLS Component Types and Identifiers

Component	Type	Identifier	Valid Values
volume serial number of a data or scratch cartridge or a cleaning cartridge	<u>vo</u>lume, <u>sc</u>ratch, <u>clean</u>	*vol_id*	Six-character identifier consisting of any combination of numbers (0-9), letters (A-Z), dollar sign ($), pound sign (#), and spaces
			Use single or double quotes to enclose volsers with leading or trailing spaces.
			Do not specify volsers with embedded spaces.
range of volume serial numbers	<u>vo</u>lume	*volrange*	Specifies an ascending range of volumes separated by a dash (-)
			If the ALPHANUM_VOL_RANGES dynamic variable is set to FALSE (default) : ACSLS supports the starting and ending vol_ids of the range that are specified as complete vol_ids (Ex: AAA000-AAA999). Specify only the right-most numeric portions of the volsers as the range. All preceding characters must be identical. *******
			If the ALPHANUM_VOL_RANGES dynamic variable is set to TRUE: ACSLS supports volume ranges consisting of any combination of numbers (0-9), letters (A-Z), dollar sign ($), pound sign (#), and spaces (leading and trailing). *******
			*** Use single or double quotes to enclose volsers in a range with leading or trailing spaces. The range must be in ascending sequence per the ACSII collating sequence.
volume media type	<u>me</u>dia	*media_type*	10-character media type identifier; can be any combination of numbers (0-9),letters (A-Z), and dashes (-). Spaces are not allowed.
volume owner	<u>ow</u>ner	*owner_id*	

Table 16. ACSLS Component Types and Identifiers

Component	Type	Identifier	Valid Values
scratch pool	<u>poo</u>l	*pool_id*	Decimal number (0-65534). Specifying an asterisk (*) for the *pool_id* reassigns the volume to its current *pool_id*.
ACSLS request	<u>re</u>quest	*request_id*	Unique decimal number (0-65535) assigned by ACSLS.
drive or volume lock	<u>loc</u>k	*lock_id*	Decimal number (0-32767)
ACSLS request	<u>re</u>quest	*request_id*	unique numeric (0-65535) request identifier assigned by ACSLS.

■ Common Command Error Messages

A common error message appears if ACSLS rejects a command because of a syntax error, invalid identifier, type, or option, process failure, database error, and so forth. For more information on common error messages, see *ACSLS Messages*.

If you enter a command that you cannot use, the following message appears:

```
Command access denied.
```

If you specify a volume that you cannot access, the following message appears:

```
Volume access denied.
```

■ audit

The <u>au</u>dit command updates the ACSLS database to match the actual inventory of library cartridges.

You must always perform an audit if:

- This is a new library

- To re-synchronize the ACSLS database with the contents of the library.

- One or more modules have been added, removed, or swapped.

- Cartridges have been manually added or removed through the door

- After library configuration settings have changed.

- If ACSLS does not know the location of one or more cartridges in the library.

- If a library has been expanded and you are auditing it for the first time

  When a SL8500 has been expanded please see the procudures, "Auditing the SL8500 After Expansion" on page 330, to audit it for the first time.

Format

audit *cap_id type identifier*...

cap_id

specifies the CAP that ACSLS uses to eject any errant volumes. You can specify a particular cap or enter an asterisk (*), which causes ACSLS to select the highest priority CAP.

You can specify the server or multiple ACSs only if CAP priority has been set in each ACS. You must specify an asterisk for the *cap_id* to allow automatic selection of a CAP in each ACS.

A single LSM audit waits until ACSLS updates the database, then reserves the CAP if required to eject errant volumes.

type identifier

specifies a library component. Table 17 lists the components you can audit.

Table 17. Valid Components for Audit

Library Component	*type*	*identifier*
all libraries	server	*none*
ACS	acs	*acs_id*
LSM	lsm	*lsm_id*
LSM panel	panel	*panel_id*
LSM subpanel	subpanel	*subpanel_id*

You can specify multiple ACSs, LSMs, panels, or subpanels in a single audit request. You cannot specify overlapping subpanels.

You can specify the server or multiple ACSs only if CAP priority has been set in each ACS. You must specify an asterisk for the *cap_id* to allow automatic selection of a CAP in each ACS. If this is not done, audit is unable to eject any invalid cartridges found.

When you audit the server or an ACS, the parent audit process creates a separate audit process for each LSM. These LSM audits run in parallel and process panel-by-panel in ascending *panel_id* sequence. To audit all LSMs in all libraries or an ACS, just audit the server or ACS. There is no need to run multiple, separate LSM audits.

However, within an LSM, regardless of the order in which you specify multiple components the audit processes these components in ascending order by

acs_id, *lsm_id panel_id*, *subpanel_id*. For example, the audit will process panel 0,0,9 before subpanel 0,0,10,1,7 even if you specify the subpanel first.

Usage

An audit updates the ACSLS database to match the actual inventory of library cartridges. You run an audit to:

- Create volume information in the database if you do not enter cartridges through the CAP. For example, run an audit if you add an LSM to your library, open the LSM door, and manually add cartridges to the LSM.

- Resolve discrepancies between the library and the database. For example, run an audit if you open an LSM door and manually remove cartridges instead of ejecting them through the CAP. The audit deletes information for the removed cartridges from the database.

- View the contents of cells specified by the audit (you must have a display monitor connected to the LSM robot's vision system).

Note: Audit recognizes particular models of cleaning cartridges and will record them in the database as cleaning cartridges. However, the audit command does not set the *max_uses* for new cleaning cartridges it encounters. You will have to set the *max_uses* for these using the set clean command on pages 160 and 300.

When audit cannot find a cartridge in any recorded location (cell and possibly drive), audit either marks the cartridge as absent (if absent volume retention is enabled) or deletes the volume. When audit deletes the volume, it removes all information about the volume from the database, including volume information and customer-supplied information about access control, scratch status, lock IDs, and pool IDs. If audit later finds the cartridge in another location, it re-adds the volume information, but the customer-supplied information is lost.

An audit ejects errant volumes and deletes their information from the database. An errant volume has:

- An external label that duplicates one already scanned.

- A missing or unreadable external label and no virtual label.

- An invalid media type.

Audits only apply to LSM storage cells, not to transports or CAPs. ACSLS displays `cmd_proc` messages during the audit and records any database changes from the audit in the event log. If volume statistics logging is enabled, ACSLS records additional volume found and volume not found messages in the `acsss_stats.log`. See "Creating a Logging Volume Statistics Report" on page 124 for more information.

Hint: Use the following guidelines for running an audit:

- The ACS or LSM being audited must be either online or in diagnostic state. Normal library processing (including mounts and dismounts) can occur during an audit, although library processing slows down the audit.

  The difference between online and diagnostic states is: an audit in diagnostic state can only be performed through the cmd_proc. Client requests are rejected while the LSM is in the diagnostic state. This state is faster because the audit now has exclusive run of the library without competing mount/dismount requests that could otherwise be submitted from the client.

- After you start an audit on an entire LSM, you cannot start another audit on the same LSM. You must cancel and rerun the audit.

- You can cancel any audit request, but ACSLS will always finish auditing the current panel or subpanel. When you cancel an audit, some or all of the cartridges marked for ejection may not be ejected. When you cancel an audit, cartridges already ejected are not re-entered.

CAUTION: If you cancel an audit or if there is a library or ACSLS hardware or software failure during the audit, you must rerun the same audit. Cartridges marked for ejection but not actually ejected during the first audit are no longer in the database and are not under ACSLS control.

Examples

To audit the entire library and specify the highest priority CAP in each ACS for ejections:

```
audit * server
```

To audit LSM 0,1 and specify CAP 0,1,1 for ejections:

```
audit 0,1,1 lsm 0,x
```

To audit panel 10 of LSM 0,1 and specify the highest priority CAP in LSM 0,1 for ejections:

```
audit 0,1,* panel 0,1,10
```

Notes

A cell cannot be audited if it is reserved by another process. If a cell is reserved, ACSLS rechecks the database until the cell becomes available up to a maximum of 60 retries. If the cell is still unavailable, the audit skips the cell and logs a message to the event log.

See Also

For information about...	See...
Guidelines for auditing the library	"Auditing the Library" on page 58
Cancelling a command	"cancel" on page 194
Displaying CAP status	"query cap" on page 233
Displaying ACSLS and library status	"query server" on page 256
Displaying ACS status	"query acs" on page 231
Displaying LSM status	"query lsm" on page 243
Setting CAP selection priority	"set cap priority" on page 262
Setting CAP mode (manual or automatic)	"set cap mode" on page 260
Changing the state of a library component	"vary" on page 277

Command Area Messages

Success Messages

The following message appears when the audit completes successfully.

```
Audit: Audit completed, Success.
```

In addition, one of the following messages appears to confirm which component was audited:

```
Audit: Audit of storage server, valid
Audit: Audit of ACS, acs_id, status valid
Audit: Audit of LSM, lsm_id, panel_id, valid
Audit: Audit of panel, panel_id, valid
Audit: Audit of subpanel, subpanel_id, valid
```

Hint: If you audit an invalid cell location (that is, just above or below a drive or on an extra column), ACSLS returns a success message to allow audits to work with PTPs.

Intermediate Messages

Intermediate messages consist of a two-line display with the following first line:

```
Audit: Intermediate response: Audit activity.
```

One of the following messages appears on the second line.

```
Audit: Volume ejected, unreadable label.
```

> **Explanation:** ACSLS ejected a cartridge that had:

- No external label

- No virtual label, or

- An unreadable label

`Audit: Volume `*`vol_id`*` will be ejected, duplicate label.`

Explanation: ACSLS ejected with a duplicate external label within the range of cells being audited.

Variable: *vol_id* is the volume with the duplicate label.

`Audit: Volume `*`vol_id`*` found.`

Explanation: The audit found a volume in the ACS that is not in the ACSLS database. The audit added the volume to the database.

Variable: *vol_id* is the volume added to the database.

`Audit: Volume `*`vol_id`*` not found.`

Explanation: A volume listed in the ACSLS database is not in the ACS. The volume is deleted from the database.

Variable: vol_id is the volume deleted from the database.

`Audit: Volume will be ejected, invalid media type`

Explanation: ACSLS ejected a volume with an invalid media type.

Error Messages

`Audit in progress.`

Explanation: ACSLS did not start the audit because another audit for the same LSM is in progress.

`CAP `*`cap_id`*` in use.`

Explanation: The CAP specified for the audit is in use.

Variable: *cap_id* is the CAP in use.

`Multiple ACS audit.`

Explanation: The audit failed because the audit command specified multiple ACSs without specifying an * for the cap_id.

`Not in same ACS.`

Explanation: The audit failed because the *cap_id* and the *identifier* specified are not in the same ACS.

Variable:

- *cap_id* is the CAP specified for the audit.

- *identifier* is the library component specified for the audit.

Display Area Messages

`cap_id Remove cartridges from CAP.`

Explanation: The audit filled the CAP with ejected cartridges. Empty the CAP, then close it to continue the audit.

Variable: *cap_id* is the CAP that contains the ejected cartridges.

`CAP cap_id: Place magazines in CAP.`

Explanation: The CAP requires magazines for the audit. Open the CAP, place magazines inside, then close the CAP.

Variable: *cap_id* is the CAP that requires magazines.

`CAP cap_id: No CAP available, waiting...`

Explanation: No CAP is available to eject cartridges.

Variable: The *cap_id* appears as it was specified in the audit command:

- *acs*, *lsm*, *cap* if the `audit` command explicitly specified the CAP.

- *acs*, *lsm*, * if the `audit` command specified the CAP as *acs*, *lsm*, *.

- *acs*, *, * if the `audit` command specified the CAP as *acs*, * or *.

■ cancel

The `cancel` command cancels a current or pending request.

Format

`cancel request_id`

Options

`request_id`
specifies the identifier of the request to cancel.

Use the `cancel` command to cancel current or pending request issued by an `audit`, `define pool`, `delete pool`, `eject`, `enter`, `lock`, `query`, `set`, or `venter` command or client application. Use the query `request` command to display the ID of the request you want to cancel.

You can cancel an audit of a server, ACS, or LSM. Because ACSLS internally translates server, ACS, or LSM audits into a series of panel audits, ACSLS completes the audit of the current panel before cancelling the remainder of the audit. You cannot cancel an audit of a panel or subpanel. When you cancel an audit, cartridges already ejected are not re-entered.

CAUTION: If you cancel an audit or if there is a library or ACSLS hardware or software failure during the audit, you must rerun the same audit. Cartridges marked for ejection but not actually ejected during the first audit are no longer in the database and are not under ACSLS control.

Hint: You must enter a `cancel` command from a different `cmd_proc` than the `cmd_proc` that issued the request you want to cancel.

The `cancel` command immediately cancels any pending requests and handles current requests as follows:

audit
>   Because ACSLS internally translates server, ACS, or LSM audits into a series of panel audits, ACSLS completes the audit of the current panel before cancelling the remainder of the audit.

CAUTION: If you cancel an audit or if there is a library or ACSLS hardware or software failure during the audit, you must rerun the same audit. Cartridges marked for ejection but not actually ejected during the first audit are no longer in the database and are not under ACSLS control.

define pool
>   ACSLS stops defining scratch pools but does not delete any scratch pools already defined.

delete pool
>   ACSLS stops deleting scratch pools but does not redefine any scratch pools already deleted.

eject
>   ACSLS stops the ejection and the cmd_proc displays a message to remove any cartridges already ejected, which are not reentered.
>
>   The eject does not terminate until the cartridges are removed, the CAP is closed, and ACSLS verfies that the CAP is empty.

enter
>   ACSLS stops the enter and, if any cartridges remain in the CAP, `cmd_proc` displays a message to remove these cartridges. Any cartridges already entered into the LSM are not ejected.
>
>   The enter does not terminate until the cartridges are removed, the CAP is closed, and ACSLS verfies that the CAP is empty.

Note: **Automatic enter**. You cannot cancel an automatic enter operation that is in progress using the cancel command. To terminate an automatic enter in progress:

-   If the CAP door is open, remove all the cartridges and close the door.

- If the CAP door is closed and the cartridges are being moved into the library, you must allow the remaining cartridges to be entered into the library. The enter then terminates.

lock

Resource locking by the specified request is stopped. If the request has not yet acquired all specified resources, none of the resources are locked.

query

ACSLS cancels the query.

set

For <u>set</u> <u>cap</u>, ACSLS stops setting CAP attributes, but does not change any attributes already set.

For <u>set</u> <u>scr</u>atch or <u>set</u> <u>clea</u>n requests, ACSLS stops setting scratch cartridge or cleaning cartridge attributes, but does not change any attributes already set.

venter

ACSLS stops the enter and, if any cartridges remain in the CAP, `cmd_proc` displays a message to remove these cartridges. Any cartridges already entered into the LSM are not ejected.

Examples

To display request IDs for all current and pending requests:

```
query request all
```

Example output of <u>query</u> <u>requ</u>est all:

```
Identifier   Command   Status

13           enter     Current
15           query     Pending
```

To cancel request 13 (current enter request) in the example above:

```
cancel 13
```

Notes

None.

See Also

For information about...	See...
Updating the ACSLS database to match the actual inventory of library cartridges	"audit" on page 188
Creating or modifying scratch pools	"define pool" on page 200
Deleting empty scratch pools	"delete pool" on page 202

For information about...	See...
Ejecting cartridges from the library	"eject" on page 207
Making a CAP (manual mode) ready to enter labelled cartridges into the library	"enter" on page 211
Locking (dedicating) drives and cartridgess to your current lock ID	"lock" on page 217
Displaying the status of a library component	"query commands" on page 230
Setting various attributes of different library components	"set commands" on page 259
Making a CAP ready to enter unlabeled cartridges into the library	"venter" on page 284

Command Area Messages

Success Messages

Request *request_id* cancelled.

> **Explanation:** ACSLS cancelled the requested command.

> **Variable:** *request_id* is the request identifier of the cancelled command.

Intermediate Messages

None.

Error Messages

`Request` *request_id* `can not be cancelled:` *status*`.`

> **Explanation:** ACSLS cannot cancel the specified command.

> **Variable:**

- *request_id* is the request identifier of the command that ACSLS cannot cancel.

- *status* is one of the following:

> `Request identifier` *request_id* `invalid.`

>> The cancel command specified an invalid request identifier.

> `Request identifier` *request_id* `not found.`

>> The cancel command specified an request identifier for a request that is not current or pending.

Display Area Messages

None.

■ clear lock

The `clear lock` command removes all active and pending locks on a specified drive or cartridge.

Format

`clear lock type` *identifier*

Options

type identifier

specifies a library component. Table 18 lists the components whose resource locks you can clear.

Table 18. Valid Components for Clear Lock

Library Component	*type*	*identifier*
transport	drive	drive_id
volume	volume	vol_id

Usage

Use the `clear lock` command to remove all active and pending locks on a specified drive or cartridge. Your current lock ID must either be 0 or it must match the lock ID of the driver or cartridge.

The `unlock` command removes only active locks on drives or cartridges. You can, however, use the `unlock` command to remove active locks on all drives or all cartridges.

Note: The `clear lock` command always resets your lock ID to 0.

Examples

To clear all locks for drive 1,1,5,2:

```
clear lock drive 1,1,5,2
```

To clear all locks for volume NN0108:

```
clear lock volume NN0108
```

Notes

You cannot cancel a clear lock command.

See Also

For information about...	See...
Locking drives and cartridges	"lock" on page 217
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Setting your lock ID	"set lock" on page 265
Displaying your lock or user ID	"show" on page 270
Removing active locks for drives or cartridges	"unlock" on page 275

Command Area Messages

Success Messages

The following message appears when a clear request succeeds:

```
Clear: Completed, Success.
```

In addition, for each identifier in the request, one of the following messages appears depending on the library component (*type*):

```
Clear: Drive drive_id all locks cleared.
Clear: Volume vol_id all locks cleared.
```

Intermediate Messages

None.

Error Messages

```
Clear: Clear lock of drive drive_id failed,
   Drive identifier drive_id available.
```

Explanation: ACSLS cannot clear locks because the specified drive is not locked.

Variable: drive_id is the identifier of the specified transport.

```
Clear: Clear lock of volume vol_id failed,
   Volume identifier vol_id available.
```

Explanation: ACSLS cannot clear locks because the specified volume is not locked.

Variable: vol_id is the identifier of the specified volume.

Display Area Messages

None.

■ define pool

The <u>def</u>ine <u>poo</u>l command creates or modifies scratch pools.

Format

<u>def</u>ine <u>poo</u>l *low_water_mark high_water_mark pool_id*...[<u>ov</u>erflow]

Options

low_water_mark
> is the low volume warning threshold. If the scratch cartridge count falls below this threshold, ACSLS logs a warning message in the event log. Valid values are 0 to 2^{31}-1. The default is 0.

high_water_mark
> is the high volume warning threshold. If the scratch cartridge count reaches or exceeds this threshold, ACSLS logs a warning message in the event log. This value must be greater than the value for *low_water_mark*.

pool_id
> specifies the pool identifier. Pool 0 is the common scratch pool, which always exists. You can modify the common scratch pool attributes.

overflow
> specifies that if this pool cannot satisfy <u>mount</u> <u>sc</u>ratch requests, ACSLS will select cartridges from the common pool (Pool 0).

Usage

Use the <u>def</u>ine <u>poo</u>l command to create or modify scratch pools.

Examples

To define a low threshold of 0, a high threshold of 600, and overflow for new pool 1:

    define pool 0 600 1 overflow

To define a low threshold of 0, a high threshold of 600, and no overflow for existing pool 5:

    define pool 0 600 5

Notes

Scratch pools are not owned by one client application or user ID. You can, however, use volume access control to restrict access for specific scratch cartridges.

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194
Deleting empty scratch pools	"delete pool" on page 202
Guidelines and procedures for managing scratch cartridges	"Managing Scratch Cartridges" on page 93
Mounting a scratch cartridge on a transport	"mount *" on page 223
Displaying scratch pool attributes	"query pool" on page 250
Setting or clearing cartridge scratch attributes	"set scratch" on page 267

Command Area Messages

Success Messages

The following message appears when a define pool command succeeds:

```
Define: Define completed, Success.
```

In addition, the following message appears for each created or modified pool:

```
Define: Pool pool_id created.
```

Intermediate Messages

None.

Error Messages

None.

Display Area Messages

```
Pool pool_id: low water mark warning.
```

Explanation: The number of volumes in the specified scratch pool is less than or equal to the low volume threshold.

Variable: `low_water_mark` is the low volume threshold of the specified scratch pool.

```
Pool pool_id: high water mark warning.
```

Explanation: The number of cartridges in the specified scratch pool is greater than or equal to the high volume threshold.

Variable: `high_water_mark` is the high volume threshold of the specified scratch pool.

■ delete pool

The <u>del</u>ete <u>poo</u>l command deletes empty scratch pools.

Format

<u>del</u>ete <u>poo</u>l *pool_id...*|<u>al</u>l

Options

pool_id
    specifies the pool ID.

    Pool 0 is the common pool, which you cannot delete.

all
    specifies all empty scratch pools.

Usage

Use the <u>del</u>ete <u>poo</u>l command to delete empty scratch pools. If a pool contains scratch cartridges, you must reassign these cartridges to another pool before deleting the first pool. If a scratch cartridge is mounted, it becomes a data cartridge but remains in its scratch pool. Use the <u>set</u> <u>scr</u>atch <u>of</u>f command to reassign the data cartridge to the common pool.

Examples

To delete all empty scratch pools:

    delete pool all

Hint: Only the empty pools will be deleted; pools with cartridges assigned will not be affected.

To delete scratch pool 1:

1. Query scratch pool 1:

    query scratch 1

```
1998-06-30 09:35:30 Scratch Status
Scratch Pool    Identifier      Home Location Status    Type
1    34813       0,0,1,8,8     home                     3480
1    34815       0,0,1,8,1     home                     3480
```

    Pool 1 has two cartridges, 34813 and 34815.

2. Empty pool 1 by reassigning its cartridges to pool 5:

    set scratch 5 348013 348015

3. Delete pool 1:

   delete pool 1

Notes

None.

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194
Creating or modifying scratch pools	"define pool" on page 200
Guidelines and procedures for managing scratch cartridges	"Managing Scratch Cartridges" on page 93
Displaying scratch pool attributes	"query pool" on page 250
Displaying the location and media type of a cartridge	"query volume" on page 257
Setting or clearing cartridge scratch attributes	"set scratch" on page 267

Command Area Messages

Success Messages

The following message appears when a <u>del</u>ete <u>poo</u>l command completes successfully:

```
Delete: Delete completed, Success.
```

In addition, the following message appears for each deleted pool.

```
Delete: Pool pool_id deleted.
```

Intermediate Messages

None

Error Messages

```
Delete: Pool pool_id failed, Pool not empty.
```

> **Explanation:** ACSLS cannot delete the scratch pool because it is not empty.

> **Variable:** *pool_id* is the identifier of the requested pool.

■ dismount

The dismount command dismounts a cartridge from a transport.

Format

dismount *vol_id drive_id* [force]

Options

vol_id
   specifies the cartridge.

drive_id
   specifies the transport.

force
   forces a dismount of the actual volume in the specified transport, even if the *vol_id* of the cartridge in the drive does not match the specified *vol_id*.

   This option also forces a dismount even if the drive is not unloaded.

Usage

Use the dismount command to dismount a volume from a drive and place the cartridge in an available storage cell.

Dismount

Use the dismount command without the force option to dismount a specified cartridge from a specified transport. A successful unforced dismount requires the following:

*   The drive must be online

*   The *vol_id* of the cartridge in the drive must match the *vol_id* you specify on the dismount command

*   The drive must be unloaded

Dismount force

Use the dismount command with the force option to force a dismount of the actual cartridge mounted in a specified transport. The *vol_id* of the cartridge in the drive does not have to match the *vol_id* you specify on the dismount command. In addition, if the cartridge is not ready for dismounting, ACSLS forces the drive to automatically rewind, unload, and dismount the cartridge. The drive must be online.

You can use the force option to dismount a cartridge with an unreadable or unknown label or a cartridge that a client application did not dismount. Note that ACSLS returns the cartridge to an available storage cell, even if the label is unreadable or missing.

CAUTION: Dismounting a cartridge with dismount force will unconditionally abort any read/write activity between the drive and any application that is currently using the drive.

This is due to the fact that a forced dismount immediately rewinds and unloads the cartridge in the drive specified.

Examples

To dismount cartridge EDU200 from drive 0,1,10,2:

```
dismount EDU200 0,1,10,2
```

Notes

None.

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194
Ejecting cartridges from the library	"eject" on page 207
Making a CAP (manual mode) ready to enter labelled cartridges into the library	"enter" on page 211
Mounting a scratch cartridge on a transport	"mount *" on page 223
Mounting a data cartridge on a transport	"mount" on page 220
Displaying CAP status	"query cap" on page 233
Displaying drive status	"query drive" on page 236
Setting CAP mode (manual or automatic)	"set cap mode" on page 260
Setting CAP selection priority	"set cap priority" on page 262

Command Area Messages

Success Messages

```
Dismount: vol_id dismounted from drive_id.
```

Explanation: An unforced dismount succeeded.

Variable:

- *vol_id* is the identifier of the dismounted cartridge.

- *drive_id* is the identifier of the specified transport.

`Dismount: Forced dismount of` *`vol_id`* `from` *`drive_id`*`.`

Explanation: A forced dismount succeeded.

Variable:

- *`vol_id`* is the identifier of the dismounted cartridge.

- *`drive_id`* is the identifier of the specified transport.

Intermediate Messages

None

Error Messages

`Dismount: Dismount failed, ACS` *`acs_id`* `full.`

Explanation: ACSLS cannot dismount the cartridge because the ACS has no free storage cells. You must eject at least one other cartridge from the ACS so that ACSLS can dismount the cartridge from the transport.

Variable: *`acs_id`* is the identifier of the ACS that contains the cartridge.

Note: Only online LSMs are used in locating free cells; therefore this message can occur even when there is an LSM with free storage cells if that LSM is offline.

`Dismount: Dismount failed, Audit in progress.`

Explanation: ACSLS cannot dismount the cartridge because an audit in progress has locked out access to the last unoccupied cell location in the ACS.

`Dismount: Dismount failed, Cartridge in drive` *`drive_id`*`, unreadable label`

Explanation: ACSLS cannot dismount the cartridge because the cartridge either has no external label, an unreadable external label, or no virtual label. Use a forced dismount to dismount the cartridge.

Variable: *`drive_id`* is the identifier of the specified transport.

`Dismount: Dismount failed, Drive identifier` *`drive_id`* `available.`

Explanation: No cartridge is mounted in the specified transport.

Variable: *`drive_id`* is the identifier of the specified transport.

```
Dismount: Dismount failed, Drive identifier drive_id in
use.
```

> **Explanation:** ACSLS cannot dismount the cartridge because the cartridge is not rewound and unloaded. Either wait for the client application must either rewind and unload the cartridge, then enter a dismount command or use a forced dismount to dismount the cartridge.
>
> **Variable:** *drive_id* is the identifier of the specified transport.

```
Dismount: Dismount failed, Misplaced tape.
```

> **Explanation:** ACSLS cannot dismount the cartridge because the external label of the cartridge does not match the cartridge identifier in the ACSLS database.   The cartridge identifier in the ACSLS database is updated to match the external label of the cartridge. Retry the dismount.

```
Dismount: Dismount failed, Cartridge not in drive.
```

> **Explanation:** ACSLS cannot dismount the cartridge because the external label of the cartridge does not match the cartridge identifier specified in the dismount. Reenter the dismount command with the correct cartridge identifier.

Display Area Messages

None.

■ eject

The eject command directs the robot to take cartridges from inside an LSM and place them into a CAP, where they can be removed by the operator.

If you select the `lsm_id` option, you can use multiple CAPs in a single LSM to eject cartridges.

Format

eject *cap_id|lsm_id vol_id|volrange...*

Options

cap_id
   specifies the CAP used to eject the cartridges.

lsm_id
   If you select the *lsm_id* option, you can use multiple CAPs that are available in a single LSM to eject cartridges. All CAPs that satisfy the requirements (such as manual, non zero priority, available) are unlocked for ejecting cartridges. You can eject cartridges through any or all the CAPs selected, and in any order. Eject initially fills the CAPs with cartridges in order of highest priority.

Example: You have two CAPs - one with priority 2 and one with priority 5; eject fills the CAP with priority 5 status first, then the CAP with priority 2 status. If there are only enough cartridges to be ejected for one CAP, then CAP with priority 5 is filled.

vol_id | volrange
specifies the external or virtual label types of the cartridge or range of cartridges to eject.

Usage

Use the eject command to eject cartridges from the library, which removes them from ACSLS control. The robot places the specified cartridge in the designated CAP, then ACSLS frees the cell locations where the cartridges were stored. Cartridge information is either retained if the ABSENT_VOLUME_RETENTION_PERIOD variable has a non-zero value, or deleted if the ABSENT_VOLUME_RETENTION_PERIOD is set to zero. If you specify more than a CAP full of cartridges on the eject command, empty the CAP when it fills, close the CAP, and continue the ejection until all cartridges are ejected.

You can use a single eject command to eject multiple cartridges not in a range by specifying multiple cartridge IDs separated by a space.

Hint: If your library has two connected LSMs and the PTP is down, to successfully eject cartridges, either:

• Specify a CAP in the LSM where the cartridge is stored. For example, if cartridge NN0100 is stored in LSM 0,0, to eject this cartridge through CAP 0,0,0:

    eject 0,0,0 NN0101

• For the *cap_id*, specify the LSM where the cartridge is stored but use a wildcard (*) for the CAP number; ACSLS will select the highest priority CAP in the LSM. For example, if cartridge NN0114 is stored in LSM 0,0, to eject this cartridge through LSM 0,0's highest priority CAP:

    `eject 0,0,* NN0114`

Examples

To eject cartridge NN0101 through CAP 0,0,0:

    eject 0,0,0 NN0101

To eject cartridge range NN0101-NN0109 through the highest priority CAP on ACS 0:

    `eject 0,* NN0101-NN0109`

To eject cartridges NN0101, NN0103, NN0105, and NN0107 through the highest priority CAP on ACS 0:

```
eject 0,* NN0101 NN0103 NN0105 NN0107
```

To use multiple CAPs for ejecting cartridges in an CAP1,2:

```
eject 1,2 RB1000-RB2000
```

Notes

Make sure you remove all ejected cartridges from the CAP. You must complete the ejection by unloading all ejected cartridges and closing the CAP door before you can use the CAP for another operation (such as an enter or audit).

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194
Dismounting a cartridge from a transport	"dismount" on page 204
Guidelines and procedures for ejecting cartridges	"Ejecting Cartridges" on page 93
Making a CAP (manual mode) ready to enter labelled cartridges into the library	"enter" on page 211
Displaying CAP status	"query cap" on page 233
Displaying drive status	"query drive" on page 236
Displaying the location and media type of a cartridge	"query volume" on page 257
Setting CAP mode (manual or automatic)	"set cap mode" on page 260
Setting CAP selection priority	"set cap priority" on page 262

Command Area Messages

Success Messages

```
Eject: Eject complete, a cartridges ejected
    Eject: vol_id ejected from cap_id
```

Explanation: ACSLS ejected the specified cartridges.

Variable:

- nn is the number of cartridges ejected.

- vol_id is the cartridge identifier of the ejected cartridge.

- cap_id is the CAP that contains the ejected cartridge.

Intermediate Messages

None

Error Messages

Eject: *vol_id* `Eject failed, CAP` *cap_id* `full.`

Explanation: A cartridge was not ejected because the CAP is full.

Variable:

- *vol_id* is the cartridge identifier of the cartridge that was not ejected.
- *cap_id* is the CAP specified for the ejection.

Eject: *vol_id* `Eject failed, CAP` *cap_id* `in use.`

Explanation: A cartridge was not ejected because the CAP is in use.

Variable:

- *vol_id* is the cartridge identifier of the cartridge that was not ejected.
- *cap_id* is the CAP specified for the ejection.

Eject: *vol_id* `Eject failed, Misplaced tape.`

Explanation: ACSLS cannot eject the cartridge because the external label of the cartridge does not match the database's cartridge identifier for the storage cell. The database updates the cartridge identifier, media type, and cleaning cartridge attribute for the cartridge in the storage cell.

Variable: *vol_id* is the cartridge identifier of the cartridge that was not ejected.

Eject: *vol_id* `Eject failed, Not in same ACS.`

Explanation: A cartridge was not ejected because it is not in the ACS specified in the *cap_id*.

Variable: *vol_id* is the cartridge identifier of the cartridge that was not ejected.

Eject: *vol_id* `Eject failed, Volume identifier` *vol_id* `not found.`

Explanation: A cartridge was not ejected because it is not:

- In the storage cell specified in the database,
- In-transit, or
- In a transport.

`ACSLS deletes the volume entry from the database.`

Variable: *vol_id* is the cartridge identifier that was deleted from the database.

`Eject:` *vol_id* `Eject failed, Cartridge in drive.`

Explanation: A cartridge was not ejected because it is mounted in a transport.

Variable: *vol_id* is the cartridge identifier of the cartridge that was not ejected.

`Eject:` *vol_id* `Eject failed, Volume` *vol_id* `in use.`

Explanation: A cartridge was not ejected because it is reserved for use by another request.

Variable: *vol_id* is the cartridge identifier of the cartridge that was not ejected.

Display Area Messages

`CAP` *cap_id* `Remove cartridges from CAP.`

Explanation: Either the CAP is full or all requested cartridges are in the CAP. Empty the CAP.

Variable: *cap_id* is the CAP that contains the ejected cartridges.

`CAP` *cap_id* `Place magazines in CAP.`

Explanation: The CAP requires magazines for the ejection. Open the CAP, place magazines inside, then close the CAP.

Variable: *cap_id* is the CAP that requires magazines.

■ enter

This command allows you to set the CAP to operate in either manual or automatic mode. For procedures for setting the CAP to automatic mode or manual mode, refer to the section "Entering Cartridges" on page 91.

- Automatic mode

    When a CAP is in automatic mode, you can initiate an enter operation without issuing an enter command. You do this by opening the CAP door, placing one or more cartridges inside, and closing the CAP. The CAP is locked while the enter is being processed; when the enter operation is compete, the CAP is unlocked.

- Manual mode

  When in manual mode, the CAP is locked and can only be used by issuing the following command prior to opening the CAP and entering the cartridges.

  ```
  enter cap_id
  ```

Options

cap_id
   specifies the CAP.

lsm_id
   If you select the *lsm_id* option, you can use multiple CAPs that are available in a single LSM to enter cartridges. All CAPs that satisfy the requirements (such as manual, non zero priority, available) are unlocked for entering cartridges. You can enter cartridges through any or all the CAPs selected, and in any order. venter is not valid and returns a CAP invalid error.

Usage

Use the <u>enter</u> command to make a manual mode CAP ready to enter labelled cartridges.

Examples

To make CAP 0,0,2 ready to enter cartridges:

enter 0,0,2

To make the highest priority CAP in LSM 0,0 ready to enter cartridges:

`enter 0,0,*`

To use multiple CAPs for entering cartridges in an LSM 1,2:

enter 1,2

Notes

Cartridges being entered into an LSM must be compatible with that LSM. For example, only T9840, T9940, and LTO cartridges can be entered into an L5500 LSM.

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194
Creating or modifying scratch pools	"define pool" on page 200
Ejecting cartridges from the library	"eject" on page 207

For information about...	See...
Guidelines and procedures for entering cartridges	"Entering Cartridges" on page 91
Mounting a scratch cartridge on a transport	"mount *" on page 223
Mounting a data cartridge on a transport	"mount" on page 220
Displaying CAP status	"query cap" on page 233
Setting CAP mode (manual or automatic)	"set cap mode" on page 260
Setting CAP selection priority	"set cap priority" on page 262
Making a CAP ready to enter unlabeled cartridges into the library	"venter" on page 284

Command Area Messages

Success Messages

After all labelled cartridges have been entered, the following message appears, with the second line of the message repeated for each cartridge successfully entered:

```
Enter: Enter complete, nn volumes entered
   Enter: vol_id Entered through cap_id
```

- *nn* is the total number of cartridges entered

- *vol_id* is the cartridge identifier of the entered cartridge

- *cap_id* is the CAP used to enter the cartridges

Intermediate Messages

None.

Error Messages

```
Enter: vol_id Enter failed, ACS acs_id full.
```

Explanation: A cartridge was not entered because the ACS has no free storage cells. You must eject at least one cartridge from the ACS so that ACSLS can enter the cartridge.

Variable:

- *vol_id* is the external label of the cartridge that was not entered.

- *acs_id* is the identifier of the ACS that has no free storage cells.

Note: ACSLS only searches online LSMs for free cells; this message can occur even if offline LSMS have free cells.

Enter: *vol_id* Enter failed, Audit in progress.

Explanation: A cartridge was not entered because an audit has locked out access to a cell location required for the enter.

Variable: *vol_id* is the external label of the cartridge that was not entered.

Enter: *vol_id* Enter failed, CAP *cap_id* in use.

Explanation: A cartridge was not entered because the specified CAP is in use for auditing, ejecting cartridges, or another enter process.

Variable:

vol_id is the external label of the cartridge that was not entered.

cap_id is the in-use CAP.

Enter: vol_id Enter failed, Duplicate label.

Explanation: A cartridge was not entered because the cartridge identifier of a cartridge in the CAP already exists in the ACSLS database.

Variable: vol_id is the external label of the cartridge that was not entered.

Enter: Enter failed, Unreadable label.

Explanation: A cartridge was not entered because the cartridge either has no external label or an unreadable external label.

Enter: *vol_id* Enter failed, Unknown media type label.

Explanation: A cartridge was not entered because the cartridge does not have a media identifier on its externals label.

Variable: *vol_id* is the external label of the cartridge that was not entered.

Display Area Messages

CAP *cap_id*: Place cartridges in CAP.

Explanation: The CAP is ready for cartridge entry. Open the CAP and enter cartridges.

Variable: *cap_id* is the CAP used to enter cartridges.

CAP *cap_id*: Remove cartridges from CAP.

Explanation: One or more cartridges cannot be entered.

Variable: *cap_id* s the CAP used to enter cartridges. Open the CAP and remove the cartridges.

CAP *cap_id*: CAP *cap_id* `Place magazines in CAP.`

> **Explanation:** The CAP uses magazines for cartridge entry. Load cartridges in the correct magazine, open the CAP, and insert the magazines.

> **Variable:** *cap_id* s the CAP used to enter cartridges.

■ idle

The `idle` command stops ACSLS from processing new requests.

Format

`idle` [`f`orce]

Hint: You *must* enter the full command name for the `idle` command. ACSLS will reject all other forms of the command, such as `i`, `id`, or `idl`.

Options

force
  forces termination of new request processing.

Usage

Use the `idle` command to stop ACSLS from processing new requests. For example, you idle ACSLS before doing maintenance or before terminating ACSLS.

Idle

Entering an `idle` command without the force option puts ACSLS in the idle-pending state. ACSLS completes current and pending requests (except pending lock requests, which are cancelled) and rejects new requests except those listed in "Notes." ACSLS then goes to idle state and does not process subsequent requests until restarted.

Idle force

Entering an **idle** command with the force option puts ACSLS in the idle state. ACSLS cancels all current and pending requests and rejects new requests except those listed in "Notes." ACSLS does not process subsequent requests until restarted. Because ACSLS does not complete current requests, forcing ACSLS to idle state may leave the database in an inconsistent state with the hardware, which requires an audit to correct.

Hint: Depending on the current requests processing when you enter an `idle force`, the event log may report process failures. If so, vary the affected LSMs offline and back online. See "Vary" on page 313 for information about the vary command.

Examples

To put ACSLS in idle-pending state:

```
idle
```

To force ACSLS to idle state:

```
idle force
```

Notes

In either idle or idle-pending state, ACSLS accepts new requests for the cancel, idle, query, start, and vary requests.

See Also

For information about...	See...
Displaying the status of a library component	"query commands" on page 230
Starting ACSLS request processing	"start" on page 271
Restarting ACSLS	"Starting ACSLS" on page 2
Stopping ACSLS	"Stopping ACSLS" on page 3
Changing the state of a library component	"vary" on page 277

Command Area Messages

Success Messages

The following message appears when ACSLS request processing stops.

```
ACSLM Request Processing Stopped: Success.
```

Intermediate Messages

None

Error Messages

```
ACSLM Request Processing Stopped: status
```

> **Explanation:** ACSLS did not stop processing requests.

> **Variable:** status is the reason for the failure. For more information on common status messages, see ACSLS Messages.

Display Area Messages

Variable:

```
Server system idle
```

Explanation: Request processing has stopped, and ACSLS is in idle state.

```
Server system idle is pending
```

Explanation: ACSLS is processing current or pending requests; idle state is pending.

■ lock

The <u>loc</u>k command locks a cartridge or drive to a lock ID that ACSLS assigns.

Format

<u>loc</u>k *type identifier*...[wait]

Options

type identifier
   specifies a library component. Table 24. lists the components that you can lock. You can specify either a drive or a cartridge but not both in a single Lock command. You can, however, use the same Lock ID for both transports and cartridges.

Table 19. Valid Components for Lock

Library Component	*type*	*identifier*
transport	<u>dr</u>ive	*drive_id*
volume	<u>vol</u>ume	*vol_id*

wait
   specifies that the lock is pending if the component is unavailable (locked or in use). ACSLS then locks the component when it is available. You can clear or cancel a pending lock request. Idling ACSLS also cancels a pending lock request.

Usage

Use the lock command to lock a cartridge or drive to a lock ID that ACSLS assigns. You can only lock available (not locked or in use) cartridges or transports.

Note: When you enter a lock command to lock a cartridge or transport, ACSLS assigns a lock ID to the cartridge or drive and then changes your lock ID to the lock ID of the cartridge or transport. You cannot use the set lock command to set your lock ID and then use the lock command to lock a cartridge or drive with the lock ID that you set with set lock.

Examples

To lock drive 0,1,10,2:

lock drive 0,1,10,2

To lock volume EDU445:

`lock volume EDU445`

Notes

None.

See Also

For information about...	See...
Removing all active or pending locks on a specified drive or cartridge	"clear lock" on page 198
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Setting your lock ID	"set lock" on page 265
Displaying your lock or user ID	"show" on page 270
Removing active locks on transports or cartridges	"unlock" on page 275

Command Area Messages

Success Messages

The following message appears when a lock request succeeds:

`Lock: Lock completed, Success.`

In addition, for each identifier in the request, one of the following messages appears:

`Lock: Drive` *drive_id* `locked under lock_id` *lock_id*`.`
`Lock: Volume` *vol_id* `locked under lock_id` *lock_id*`.`

Where:

- *drive_id* is the locked transport.

- The *vol_id* is the cartridge identifier of the locked cartridge.

- *lock_id* is the lock ID.

Intermediate Messages

None.

Error Messages

One of the following messages appears if the lock request fails:

`Lock: Lock of drive` *`drive_id`* `failed, Drive in use.`

Explanation: ACSLS cannot lock the specified drive because the drive is already locked or in use.

Variable: *`drive_id`* is the drive that ACSLS cannot lock.

`Lock: Lock of drive` *`drive_id`* `failed, Lock failed.`

Explanation: ACSLS cannot lock the specified transport. Reenter the lock command with the correct syntax and drive identifier.

Variable: drive_id is the drive that ACSLS cannot lock.

`Lock: Lock of volume vol_id failed, Volume in use.`

Explanation: ACSLS cannot lock the specified cartridge because the cartridge is already locked or in use.

Variable: vol_id is the cartridge that ACSLS cannot lock.

`Lock: Lock of drive vol_id failed, Lock failed.`

Explanation: ACSLS cannot lock the specified cartridge. Reenter the lock command with the correct syntax and cartridge identifier.

Variable: vol_id is the cartridge that ACSLS cannot lock.

Display Area Messages

None.

■ logoff

The logoff command exits from a `cmd_proc`.

Format

`logoff`

Options

None.

Usage

Use the logoff command to exit from `cmd_proc`. If you are running an interactive (window) `cmd_proc`, logoff also quits the `cmd_proc` window.

Examples

To exit a `cmd_proc`:

logoff

`Notes`

`logoff` exits from `cmd_proc` only, is valid in any ACSLS state, and does not affect ACSLS operations.

See Also

For information about...	See...
Starting a cmd_proc	"Starting a cmd_proc" on page 6
Starting ACSLS	"Starting ACSLS" on page 2
Stopping ACSLS	"Stopping ACSLS" on page 3
Using a cmd_proc	"Using a cmd_proc" on page 4

Command Area Messages

None.

Display Area Messages

None.

■ mount

The `mount` command mounts a data cartridge.

Format

`mount` *vol_id drive_id* [`by`pass] [`rea`donly]

Options

vol_id
 specifies the cartridge.

drive_id
 specifies the transport.

bypass
 overrides some of the checks that ACSLS does before trying to mount a cartridge. For more information, see "mount bypass" on page 221.

readonly
 specifies that the cartridge is mounted write-protected.

CAUTION: LTO drives do not support mounts with write protect. If a "mount read-only" is attempted to an LTO drive, it will fail with a "Drive cannot honor write protect" message in the Event Log.

Also, some early 97xx SCSI-attached libraries do not support mounts with write protect. Even if you specify the read-only option on a mount command, the drive may write to the

cartridge. To write protect cartridges for these transports, you must use the cartridge's readonly protection (such as a thumbwheel).

Usage

Use the `mount` command to mount a data cartridge. You can only mount one cartridge on one drive each time you enter a `mount` command.

A successful mount requires the following:

* The cartridge and drive must be in the same ACS.

* The cartridge must be available and the drive must be online and available.

mount bypass

The `bypass` option overrides the checks that ACSLS does before trying to mount a cartridge as follows:.

* The `bypass` option always overrides the ACSLS verification of the cartridge ID on the external label.

* The `bypass` option overrides the ACSLS verification of compatibility between the drive type and cartridge media. Use this option only when you know that the drive and cartridge media are, in fact, compatible. For example, you could use the bypass option to force a mount of a 9840 cartridge in a T9840 drive (in a a 9310) if the cartridge's external label is missing the 7th (media type) character.

CAUTION: Do not use the bypass option unless you know that the media type of the specified cartridge is compatible with the transport. Otherwise you can damage the cartridge, the transport, or both.

Note: Although the bypass option overrides ACSLS verification of media and tape drive compatibility, some libraries also enforce media drive compatibility (e.g. SL8500 and L700). These libraries will still fail a mount of incompatible media even though an ACSLS mount bypass was issued.

Examples

To mount volume EDU010 on drive 0,0,10,2:

mount EDU010 0,0,10,2

In the following example, YUMA15 is a DD3C volume whose cartridge label is missing the seventh character. The bypass option bypasses the media compatibility checks and forces a mount of the cartridge on drive 0,0,4,0, an SD3 transport.

To mount YUMA15 on drive 0,0,4,0 using the bypass option:

```
mount YUMA15 0,0,4,0 bypass
```

Notes

None.

See Also

For information about...	See...
Dismounting a cartridge from a transport	"dismount" on page 204
Making a CAP (manual mode) ready to enter labelled cartridges into the library	"enter" on page 211
Displaying CAP status	"query cap" on page 233
Displaying drive status	"query drive" on page 236
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Displaying the status of media-compatible transports for a specified data cartridge	"query mount" on page 245
Displaying the location and media type of a cartridge	"query volume" on page 257
Setting CAP mode (manual or automatic)	"set cap mode" on page 260
Setting CAP selection priority	"set cap priority" on page 262
Making a CAP ready to enter unlabeled cartridges into the library	"venter" on page 284

Command Area Messages

Success Messages

```
Mount: vol_id mounted on drive_id
```

Explanation: ACSLS mounted the specified cartridge.

Variable:

- *vol_id* is the cartridge identifier of the cartridge that ACSLS mounted.

- *drive_id* is the drive that mounted the cartridge.

Intermediate Messages

None.

Error Messages

`Mount: Mount failed, Audit in progress.`

Explanation: ACSLS cannot mount the cartridge because an audit has locked out access to the cell location of the specified cartridge.

`Mount: Mount failed, In use.`

Explanation: ACSLS cannot mount the cartridge because either the drive is in use or the requested cartridge is reserved for another command.

`Mount: Mount failed, Misplaced tape.`

Explanation: ACSLS cannot mount the cartridge because the external label of the cartridge does not match the database's cartridge identifier for the storage cell. The database updates the cartridge identifier, media type, and cleaning cartridge attribute for the cartridge in the storage cell.

`Mount: Mount failed, Not in same ACS.`

Explanation: ACSLS cannot mount the cartridge because the specified cartridge and drive are not in the same ACS.

`Mount: Mount failed, Cartridge in drive.`

Explanation: ACSLS cannot mount the cartridge because the specified cartridge is already mounted in a transport.

`Mount: Mount failed, Unreadable label.`

Explanation: ACSLS cannot mount the cartridge because the cartridge has no label, an unreadable label, or no virtual label.

`Mount: Mount failed, Invalid media type.`

Explanation: ACSLS cannot mount the cartridge because the specified cartridge's media type is invalid.

`Mount: Mount failed, Invalid drive type.`

Explanation: ACSLS cannot mount the cartridge because the specified transport's type is invalid.

`Mount: Mount failed, Incompatible media type.`

Explanation: ACSLS cannot mount the cartridge because the cartridge's media type is incompatible with the specified transport.

Display Area Messages

None.

■ mount *

The *mount * command selects a scratch cartridge and mounts it.

Format

<u>mou</u>nt * *drive_id* [*pool_id*] [<u>me</u>dia *media_type* | <u>me</u>dia *]

Options

drive_id
> specifies the transport.

pool_id
> specifies the pool from which ACSLS selects the scratch cartridge.
> *pool_id* is optional; if you do not specify a *pool_id* ACSLS tries to find
> a scratch cartridge in the common pool (pool 0).
>
> If you specify a *pool_id* and the pool does not contain a scratch cartridge
> (or one of the correct media for mixed-media libraries) if the pool is set for
> overflow; ACSLS tries to find a scratch cartridge in the common pool
> (pool 0).

media *media_type* | <u>me</u>dia *
> specifies the cartridge media type. Specifying the media type is optional;
> see "Usage" for more information.

Usage

Use the <u>mou</u>nt <u>*</u> command to select a scratch cartridge and mount it. The
following sections describe how ACSLS selects the scratch cartridge to be
mounted and how to specify the cartridge media type for a scratch cartridge.

How ACSLS Selects Scratch Cartridges

The <u>mou</u>nt <u>*</u> command selects a scratch cartridge by:

- Creating a list of LSMs in the ACS, based on proximity to the LSM
  containing the specified transport.

- Examining each LSM in the list, until it finds scratch cartridges that meet
  the pool and media type criteria.

- Choosing the scratch cartridge with the least recent access date in that
  LSM.

The selected scratch cartridge is then mounted on the transport.

Specifying the Media Type

You can specify cartridge media type in any of the following ways:

- Explicitly specify the media type in this form of the mount * command:

  mount * *drive_id* [*pool_id*] media *media_type*

  Example: To mount a 9940 scratch cartridge from the common pool
  (pool 0):

      mount * 0,0,10,2 media STK2P

- Use a wildcard (*) for the media type so that ACSLS selects the media by scratch preferences; for more information see "Setting Scratch Preferences" on page 66. To use scratch preferences, enter this form of the `mount  *` command:

`mount  *  `*`drive_id`*` [`*`pool_id`*`] media *`

Example: To use scratch preferences to mount a scratch cartridge from the common pool (pool 0):

mount * 0,0,10,2 media *

- Omit the media option in this form of the `mount  *` command. This instructs ACSLS to pick a media type compatible with the transport.

`mount  *  `*`drive_id`*` [`*`pool_id`*`]`

Example: To mount a scratch cartridge compatible with the drive from the common pool (pool 0):

mount * 0,0,10,2

Examples

The following sections show mount scratch examples for single-media and mixed-media libraries.

Single-Media Libraries

To mount a scratch cartridge from pool 5 on drive 0,0,10,2:

`mount * 0,0,10,2 5`

> **Hint:** If no cartridge is available from pool 5 and it has been set for `overflow` ACSLS will select a cartridge from the common pool (pool 0).

To mount a scratch cartridge from the common pool (pool 0) on drive 0,0,10,0

`mount * 0,0,10,0`

Mixed-Media Libraries

To mount a scratch cartridge from pool 5 with a media type of 3480 on drive 0,0,10,2:

mount * 0,0,10,2 5 media 3480

> **Hint:** If no cartridge is available from pool 5 and it has been set for `overflow` ACSLS will select a cartridge with the specified media type from the common pool (pool 0).

To mount a scratch cartridge from pool 10 with media type determined by scratch preferences on drive 0,0,2,3:

`mount * 0,0,2,3 10 media *`

> **Hint:** If no cartridge is available from pool 10 and it has been set for `overflow` ACSLS will select a cartridge with the specified media type from the common pool (pool 0).

To mount a scratch cartridge from the common pool (pool 0) with media type of 3480 on drive 0,0,10,2:

```
mount * 0,0,10,2 media 3480
```

To mount a scratch cartridge from the common pool (pool 0) with media type determined by scratch preferences on drive 0,0,2,3:

```
mount * 0,0,2,3 media *
```

To mount a scratch cartridge from the common pool (pool 0) with media compatible with drive 0,0,2,3:

mount * 0,0,2,3

Notes

None.

See Also

For information about...	See...
Creating or modifying scratch pools	"define pool" on page 200
Deleting empty scratch pools	"delete pool" on page 202
Dismounting a cartridge from a transport	"dismount" on page 204
Managing scratch cartridges	"Managing Scratch Cartridges" on page 93
Displaying drive status	"query drive" on page 236
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Displaying the status of media-compatible transports for a specified scratch pool	"query mount *" on page 247
Displaying scratch pool attributes	"query pool" on page 250
Displaying scratch cartridge status	"query scratch" on page 254
Displaying the location and media type of a cartridge	"query volume" on page 257
Setting or clearing cartridge scratch attributes	"set scratch" on page 267
Setting scratch preferences	"Setting Scratch Preferences" on page 66

Command Area Messages

Success Messages

Mount: *vol_id* mounted on *drive_id*

> **Explanation:** ACSLS mounted the specified cartridge.

> **Variable:**

> - *vol_id* is the cartridge identifier of the cartridge that ACSLS mounted.

> - *drive_id* is the drive that mounted the cartridge.

Intermediate Messages

None.

Error Messages

Mount: Mount failed, Audit in progress.

> **Explanation:** ACSLS cannot mount the cartridge because an audit has locked out access to the cell location of the specified cartridge.

Mount: Mount failed, In use.

> **Explanation:** ACSLS cannot mount the cartridge because either the drive is in use or the requested cartridge is reserved for another command.

Mount: Mount failed, Misplaced tape.

> **Explanation:** ACSLS cannot mount the cartridge because the external label of the cartridge does not match the database's cartridge identifier for the storage cell. The database updates the cartridge identifier, media type, and cleaning cartridge attribute for the cartridge in the storage cell.

Mount: Mount failed, Not in same ACS.

> **Explanation:** ACSLS cannot mount the cartridge because the specified cartridge and drive are not in the same ACS.

Mount: Mount failed, Cartridge in drive.

> **Explanation:** ACSLS cannot mount the cartridge because the specified cartridge is already mounted in a transport.

Mount: Mount failed, Unreadable label.

> **Explanation:** ACSLS cannot mount the cartridge because the cartridge has no label, an unreadable label, or no virtual label.

Mount: Mount failed, Invalid media type.

> **Explanation:** ACSLS cannot mount the cartridge because the specified cartridge's media type is invalid.

`Mount: Mount failed, Invalid drive type.`

Explanation: ACSLS cannot mount the cartridge because the specified transport's type is invalid.

`Mount: Mount failed, Incompatible media type.`

Explanation: ACSLS cannot mount the cartridge because the cartridge's media type is incompatible with the specified transport.

`Mount: Mount failed, No compatible scratch cartridges in pool.`

Explanation: ACSLS cannot mount the cartridge because there are no scratch cartridges in the ACS of the specified drive that match the scratch cartridge's media type. In addition, if the overflow attribute is set for the pool, there are no scratch cartridges with valid media types.

Display Area Messages

`Pool `*`pool_id`*`: low water mark warning.`

Explanation: The number of cartridges in the specified scratch pool is less than or equal to the low cartridge threshold.

Variable: *`low_water_mark`* is the low threshold of the specified scratch pool.

`Pool `*`pool_id`*`: high water mark warning.`

Explanation: The number of cartridges in the specified scratch pool is greater than or equal to the high cartridge threshold.

Variable: *`high_water_mark`* is the high water cartridge threshold of the specified scratch pool.

■ move

The move command moves a specified cartridge to an available storage cell in a specified LSM.

Format

`move vol_id lsm_id`

Options

`vol_id`
specifies the cartridge.

`lsm_id`
specifies the LSM that will contain the moved cartridge.

Usage

Use the <u>mov</u>e command to move a specified cartridge to an available storage cell in either:

*   A different panel in the same LSM. For example, if you want to empty an entire panel in an LSM, you can move all the cartridges in that panel to a different locations in the same LSM.

*   A different LSM.

You can move only one cartridge each time you enter the move command. If you specify the LSM where the cartridge currently resides, ACSLS will move the cartridge to another panel within that LSM. Otherwise, ACSLS will move it to the LSM you specify. You cannot cancel a <u>mov</u>e command.

A successful move requires the following:

*   The cartridge must be available and in the same ACS as the specified LSM.

*   The LSM where the cartridge currently resides and the specified LSM must both be online. The specified LSM must have at least one available storage cell. If the move is within the same LSM, there must be at least one available cell in a different panel in that LSM. If the move requires a pass-thru, any LSMs used must also be online.

Examples

To move cartridge EDU010 (which resides in LSM 0,1) to a different panel in this LSM:

```
move EDU010 0,1
```

To move cartridge EDU010 (which resides in LSM 0,1) to LSM 0,2:

```
move EDU010 0,2
```

Notes

None.

See Also

For information about...	See...
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Displaying the status of an LSM	"query lsm" on page 243
Displaying the location and media type of a cartridge	"query volume" on page 257

Command Area Messages

Success Messages

```
Move: vol_id moved to location cell_id
```

Explanation: ACSLS moved the specified cartridge to the specified cell location.

Variable:

- `vol_id` is the cartridge identifier of the cartridge that ACSLS moved.

- `cell_id` is the new cell location of the specified cartridge.

Intermediate Messages

None.

Error Messages

```
Move: Move failed,
```

Explanation:

Variable:

Display Area Messages

None.

■ query commands

The `query` commands display the status of a library component. See the following sections for more information about each query command, including format, options, and usage.

Format

The following shows the general format of the query commands:

```
query type [subtype | *] identifier... | all
```

Notes

If a `cancel` request is issued against a pending or current query request, the display of information is halted.

See Also

For information about...	See...
Cancelling a command	*"cancel" on page 194*
Display Command	*"display Commands" on page 288*

Command Area Messages

Success Messages

Note: If the query succeeds, the requested status appears.

Intermediate Messages

None.

Error Messages

`Library not available.`

> **Explanation:** The query failed because ACSLS is recovering (all queries *except* `query server`).

`Cartridge identifier vol_id not found.`

> **Explanation:** The `query mount` *command cannot display status for the specified cartridge because it is not in the library.
>
> **Variable:** *vol_id* is the specified cartridge.

`Invalid media type`

> **Explanation:** The `query mount` * command cannot display status for the specified cartridge because it either has an invalid media type or you specified an invalid media type on the command.
>
> **Variable:** `vol_id` is the specified cartridge.

Display Area Messages

None.

■ query acs

The `query acs` command displays ACS status.

Format

`query acs acs_id... | all`

Options

`acs_id | `<u>`all`</u>

    specifies the ACS to query or all for all ACSs.

Usage

Use the <u>`query`</u> <u>`ac`</u>`s` command to display the status of an ACS in the following format:

```
yyy-mm-ddhh:mm:ssACS Status
 Identifier  State  Free Cell  Audit  Mount  Dismount  Enter  Eject
                    Count      C/P    C/P    C/P       C/P    C/P

 acs_id      state  count      n/n    n/n    n/n       n/n    n/n
```

Where:

`acs_id`

    is the ACS identifier.

`state` is one of the following ACS states:

`diagnostic`

    The ACS will process only current and pending requests and will reject any new requests. The ACS is unavailable to client applications and can only be controlled via cmd_proc. Use the vary command to vary the ACS online.

`offline`

    The ACS is offline.

`offline pending`

    The ACS will process current and pending requests, then go offline. The ACS will reject any new requests.

`online`

    The ACS is online.

`recovery`

    The ACS is initializing or recovering from errors. Wait for the ACS to go online.

`count`

    is the number of free cells in the ACS.

`n`

    is the number of current (C) and pending (P) requests for the ACS for each command that requires library resources (audit, mount, dismount, enter, and eject).

Examples

To query ACS 1:

```
query acs 1
```

To query all ACSs in the library:

```
query acs all
```

See Also

For information about...	See...
Changing the state of a library component	"vary" on page 277
Displaying request status	"query request" on page 253

■ query cap

The query cap command displays CAP status.

Format

query cap *cap_id*... | all

Options

cap_id | all
   specifies the CAP to query or all for all CAPs.

Note: You cannot specify a *cap_id* that contains asterisks.

Usage

Use the query cap command to display the status of a CAP.

The query cap command displays CAP status in the following format:

```
yyy-mm-dd    hh:mm:ss   CAP Status
Identifier   Priority      Size        State      Mode       Status

cap_id        cap_priority cap_size    cap_state cap_mode status
```

Where:

cap_id
   is the CAP identifier.

cap_priority
   is the CAP priority.

cap_size
   is the number of cells in the CAP.

cap_state
   is one of the following CAP states:

online
   The CAP is online.

offline
   The CAP is offline.

offline-pending
   The CAP will process current and pending requests, then go offline. The CAP will reject any new requests.

diagnostic
   The CAP will process only current and pending requests and will reject any new requests. The CAP is unavailable to client applications and can only be controlled via `cmd_proc`. Use the <u>va</u>ry command to vary the CAP online.

recovery
   The CAP is initializing or recovering from errors. Wait for the CAP to go online.

cap_mode
   is one of the following CAP entry modes:

`manual`
   You must unlock the CAP before entering cartridges.

`automatic`
   The CAP is ready for cartridge entry.

status
   is one of the following CAP statuses:

`available`
   The CAP is available.

<u>en</u>ter
   The CAP is unavailable (reserved for entering cartridges).

<u>ej</u>ect
   The CAP is unavailable (reserved for ejecting cartridges).

<u>au</u>dit
   The CAP is unavailable (reserved for audit processing).

Examples

To query CAP 0,1,0:

```
query cap 0,1,0
```

To query all CAPs in the library:

```
query cap all
```

See Also

For information about...	See...
Displaying request status	"query request" on page 253
Setting CAP mode (manual or automatic)	"set cap mode" on page 260
Setting CAP selection priority	"set cap priority" on page 262
Changing the state of a library component	"vary" on page 277

■ query clean

The `query clean` command displays cleaning cartridge status. Absent or ejected cartridges are not displayed.

Format

```
query clean vol_id... | all
```

Options

vol_id | all
   specifies the cleaning cartridge to query or all for all cartridges.

Usage

Use the `query clean` command to display the status of a cleaning cartridge in the following format:

```
yyy-mm-ddhh:mm:ssCleaning volume Status
 Identifier  Home Location  Max Usage  Current Use  Status  Type

vol_id      cell_id        max_usage  current_use  status  type
```

Where:

vol_id
   is the cartridge identifier of the cleaning cartridge.

cell_id
   is the location of the cleaning cartridge.

max_usage
   is the number of times the cleaning cartridge can be used.

current_usage
   is the number of times the cleaning cartridge has been used.

status
is the location of the cleaning cartridge:

home
   The cartridge is in a storage cell.

in drive
   The cartridge is in a transport.

in transit
   The cartridge is being moved.

type is the cartridge media type (for example, 3480, DD3D, DLTIII, or STK1R).

Examples

To display status information for cleaning cartridge J35992:

    query clean J35992

To query all cleaning cartridges:

    query clean all

See Also

For information about...	See...
Guidelines and procedures for cleaning transports	"Cleaning Cartridges" on page 97
Setting cleaning cartridge attributes	"set clean" on page 264

■ query drive

The query drive command displays drive status.

Format

query drive *drive_id*... | all

Options

`drive_id | all`
    specifies the drive to query or all for all transports.

Usage

Use the `query drive` command to display the status of a drive in the following format:

```
yyy-mm-ddhh:mm:ssDrive Status
Identifier   State   Status   volume   Type

drive_id     state   status   vol_id   type
```

Where:

`drive_id`
    is the drive identifier.

`state`
    is one of the following:

`online`
    The drive is online.

`offline`
    The drive is offline.

`diagnostic`
    The drive will process only current and pending requests and will reject any new requests. The drive is unavailable to client applications and can only be controlled via `cmd_proc`. Use the `vary` command to vary the drive online.

`recovery`
    The drive is initializing or recovering from errors. Wait for the drive to go online.

`status`
    is one of the following drive statuses:

`In use`
    The drive has a cartridge mounted or is reserved for a mount.

    Possible Scenario: You perform a *query drive all* and get a message saying the drive is in use. You then perform a *display drive ** and get a message saying the drive is reserved. This means: the reserved status for the drive implies that a mount request is in progress and that a

cartridge is on its way to the drive. At the same time the drive is considered to be in use.

`Available`
The drive is available for a mount.

vol_id
is the identifier of the cartridge in the transport.   This field is blank if there is no cartridge in the drive or if the cartridge's external label is unreadable or unknown.

drive_type
is the drive type.

Examples

To query drive 0,3,1,0:

query drive 0,3,1,0

To query all transports:

query drive all

See Also

For information about...	See...
Changing the state of a library component	"vary" on page 277
Dismounting a cartridge from a transport	"dismount" on page 204
Drive serial numbers	"display Commands" on page 288 and "display drive" on page 294

■ query lmu

The `query lmu` command displays LMU and port status for both single-LMU and dual-LMU ACS configurations, as well as the desired state for ACSs and ports. If a library is partitioned, it also displays the partition ID.

Note: ACSLS supports dual-LMU configurations for only the 9330 LMU with host/LMU microcode compatibility Level 12. The same microcode level must be loaded in both LMUs.

Format

query lmu *acs_id*... | all

Options

acs_id | all

specifies the ACS whose LMUs you want to query or all to query LMUs for all ACSs.

Usage

Use the `query lmu` command to display LMU and port status, as well as desired state, for both single-LMU and dual-LMU ACS configurations in the following format:

```
yyyy-mm-ddhh:mm:ssLMU Status

ACS: acs_id     Mode: mode Master Status: status
partition_status              Standby Status: status

    ACS State     Desired State
    acs_state     acs_desired_state

Port    Port State  Desired State    Role  CL          Port Name
port_id port_state  desired_state    role  compat_lvl    dev_name
```

Output Example:

```
ACSSA> q lmu 0
2008-02-27 06:08:02

ACS:   0      Mode: Single LMU         Master Status:  Not
Comm
Not Partitioned                        Standby Status:  -

      ACS State     Desired State
      offline       online

Port    Port State  Desired State    Role  CL  Port Name
0, 0    offline      online           -     13   172.27.2.6
```

Where:

acs_id
  is the ACS identifier.

mode
  is the LMU mode (Dual LMU, Single LMU, or SCSI LMU).

status
  is the master or standby LMU status (Communicating, Not Communicating, or Offline).

partition_status
>   shows if the library is partitioned. If it is partitioned it shows the partition to which you are connected. The partition status is:

>   `Not Partitioned`

>   `Partition 1-n`
>> The partiiton number.

acs_state
>   is the actual ACS state. States are:

>   `online`
>> The ACS is online.

>   `diagnostic`
>> The ACS will process only current and pending requests and will reject any new requests. The ACS is unavailable to client applications and can only be controlled via `cmd_proc`. Use the vary command to vary the ACS online.

>   `offline`
>> The ACS is offline.

>   `offline pending`
>> The ACS will process current and pending requests, then go offline. The ACS will reject any new requests.

acs_desired_state
>   you want the ACS to be in this state. The desired states are:

>   `online`
>   `diagnostic`
>   `offline`

port_id
>   is the port identifier.

port_state
>   is one of the following actual port states:

>   `online`
>> The port is online.

>   `offline`
>> The port is offline.

port_desired_state
>   you want to port to be in this state. The desired states are:

>   `online`
>   `offline`

role (des)
>   is the LMU's role and designation (A or B) where roles are:

Master
>   LMU is in master role (the LMU is managing the ACS).

Standby
>   LMU is in standby role (not managing the ACS, communicating with the master LMU and available for switchover).

Note: During switchover, the *role* field information is not current, and may display as a single dash (-). As the information becomes current, ACSLS refreshes the *role* field with the actual role of each LMU.

compat_level
>   is the host/LMU microcode compatibility level. Level 11 or above is required for dual-LMU configurations.

dev_name
>   is the port device name.

Examples

To display LMU and port status for all LMUs managing all ACSs:

    query lmu all

To display LMU and port status for all LMUs managing ACSs 0 and 1:

query lmu 0 1

See Also

For information about...	See...
Manually switching ACS management from the ACS's master LMU to the standby LMU	"switch lmu" on page 272

■ query lock

The query lock command displays the lock status of a drive or cartridge.

Format

query lock *type identifier*... | all

Options

type identifier | all

specifies the drive or cartridge to query or all for all transports or cartridges as shown in Table 20.

Table 20. Valid Lock Types for query lock

Library Component	*type*	*identifier*
transport	<u>dr</u>ive	drive_id
volume	<u>vol</u>ume	vol_id

Usage

Use the <u>q</u>uery <u>loc</u>k command to display the lock status of a drive or cartridge in the following format:

```
yyy-mm-ddhh:mm:ssLock Status
Identifier   Lock-id   Duration   Pending   Status   User Identifier

vol_id       lock_id   duration   pending   status   user_id
```
or
```
drive_id     lock_id   duration   pending   status   user_id
```

Where:

vol_id
   is the identifier of the specified cartridge.

drive_id
   is the identifier of the specified transport.

lock_id
   is the lock ID.

duration
   is the amount of time, in seconds, that the lock has been active.

pending
   is the number of lock requests that are waiting for cartridge or transport.

status is one of the following statuses:

   available
      The cartridge or drive is available.

   in use
      The cartridge or drive is in use or is reserved for a mount.

   user_id

is the user ID that has locked the cartridge or transport. The *user_id* wraps after 80 characters.

Examples

To display lock status information for drive 1,0,4,0:

q loc dr 1,0,4,0

To display lock status information for all transports:

query lock drive all

To display lock status information for cartridge SL4493:

query lock cartridge SL4493

To display lock status information for all cartridges:

query lock cartridge all

See Also

For information about...	See...
Removing all active or pending locks on a specified drive or cartridge	"clear lock" on page 198
Locking transports and cartridges	"lock" on page 217
Setting your lock ID	"set lock" on page 265
Displaying your lock or user ID	"show" on page 270
Removing active locks	"unlock" on page 275

■ query lsm

The query lsm command displays LSM status.

Format

query lsm *lsm_id*... | all

Options

lsm_id | all
    specifies the LSM to query or all for all locks.

Use the query lsm command to display the status of an LSM in the following format:

```
yyy-mm-ddhh:mm:ssLSM Status
Identifier   State    Free Cell    Audit    Mount    Dismount    Enter    Eject
                      Count        C/P      C/P      C/P         C/P      C/P

lsm_id       state    count        n/n      n/n      n/n         n/n      n/n
```

Where:

lsm_id
   is the LSM identifier.

state is one of the following LSM states:

   diagnostic
      The LSM will process only current and pending requests and will reject any new requests. The LSM is unavailable to client applications and can only be controlled via cmd_proc. Use the vary command to vary the LSM online.

   offline
      The LSM is offline.

   offline pending
      The LSM will process current and pending requests, then go offline. The LSM will reject any new requests.

   online
      The LSM is online.

   recovery
      The LSM is initializing or recovering from errors. Wait for the LSM to go online.

count
   is the number of free storage cells in the LSM.

n

   is the number of current (C) and pending (P) requests for the LSM for each command that requires library resources (audit, mount, dismount, enter, and eject).

Examples

To display status information for LSM 1 of ACS 0:

```
query lsm 0,1
```

To display status information for all LSMs:

query lsm all

See Also

For information about...	See...
Displaying request status	"query request" on page 253
Changing the state of a library component	"vary" on page 277

■ query mount

The `query mount` command displays the status of media-compatible drives for a specified data cartridge. These drives are not displayed if a cartridge(s) is absent or rejected. In addition:

- When selecting the list of drives to be returned to a `query mount` request, the drives must be compatible with the volume specified.

- Drives are ordered primarily by pass-thru distance. The drives in LSMs closest to the cartridge are listed first.

- Within a given pass-thru distance, the least-recently-used drive is returned first and the most recently used drive is returned last. This causes drives to be used evenly within a given pass-thru distance.

Format

`query mount` *vol_id*

Options

vol_id specifies the cartridge to query.

Usage

Use the `query mount` command to display the status of all library transports attached to the same ACS as the cartridge and compatible with the media type of the specified cartridge. `query mount` displays drive status for LSMs in any state (online, offline, offline pending, or diagnostic).

The compatible transports are ordered by proximity to the specified cartridge in the following format:

```
yyy-mm-dd  hh:mm:ss  Mount Status
Identifier   Status   Drive      State   Status        Volume    Drive Type

vol_id       vol_stat drive_id   state   drive_stat    inu_id    drive_type
```

Where:

vol_id
  is the identifier of the specified cartridge.

vol_stat
  is the location of the cartridge:

  home
    The cartridge is in a storage cell.

  in drive
    The cartridge is in a transport.

  in transit
    The cartridge is being moved or is missing.

drive_id
  is a list of all library transports compatible with the media type of the specified cartridge.

drive_id
  is the drive identifier.

state
  is one of the following drive drive states:

  online
    The drive is online.

  offline
    The drive is offline.

  diagnostic
    The drive will process only current and pending requests and will reject any new requests. The drive is unavailable to client applications and can only be controlled via cmd_proc. Use the vary command to vary the drive online.

  recovery

The drive is initializing or recovering from errors. Wait for the drive to go online.

status
>   is one of the following drive statuses:

>   `In use`
>>   The drive has a cartridge mounted or is reserved for a mount.

>   `Available`
>>   The drive is available for a mount.

inu_id
>   is the identifier of the cartridge in the transport. The cartridge id appears only if the *drive_stat* is In use.

drive_type
>   is the drive type.

Examples

To display status information of transports ordered by proximity to cartridge ZUNI14:

```
query mount ZUNI14
```

See Also

For information about...	See...
Mounting a data cartridge on a transport	"mount" on page 220
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Displaying the location and media type of a cartridge	"query volume" on page 257
Changing the state of a library component	"vary" on page 277

■ query mount *

When selecting the list of drives to be returned with a `query mount` request, the drives must be compatible with the media type of the volume specified.

The `query mount` command returns drives with the oldest dismount time. The drive with the oldest dismount time is considered the least-recently-used drive and the compatible drives are listed (returned from the query) in the following order:

- Drives are ordered primarily by pass-thru distance. The drives in LSMs closest to the cartridge are listed first.

- Within a given pass-thru distance, the least-recently-used drive is returned first and the most recently used drive is returned last. This causes drives to be used evenly within a given pass-thru distance.

The `query mount *` command displays the status of media-compatible drives for a specified scratch pool (and, optionally, for a specific cartridge media type within the pool).

Format

query mount * *pool_id*... [media *media_type* | media *]

Options

pool_id
    specifies the scratch pool to query.

media *media_type* | media *
    specifies the media type.

Usage

Use the `query mount *` command to display the status of all library transports compatible with all cartridge media types in a specified scratch pool and in the same ACS as the cartridges. Pool 0 is the common scratch pool. Specify the *media_type* option to restrict the display to transports compatible with a specific cartridge media type within the pool. The displayed transports are ordered by proximity to the densest scratch pools. `query mount *` displays drive status for LSMs in any state (online, offline, offline-pending, or diagnostic).

The `query mount *` command displays drive status in the following format:

```
yyy-mm-dd  hh:mm:ss  Mount Scratch Status
Identifier      Drive     State     Volume     Status       Drive Type

pool_id         drive_id  state     vol_id     drive_stat   drive_type
```

Where:

pool_id
    is the specified scratch pool.

drive_id
    is a list of all library transports compatible with all media types of the specified pool (or a specific media type within the pool if a media type was specified).

state

is one of the following drive states:

`online`
> The drive is online.

`offline`
> The drive is offline.

`diagnostic`
> The drive will process only current and pending requests and will reject any new requests. The drive is unavailable to client applications and can only be controlled via `cmd_proc`. Use the vary command to vary the drive online.

`recovery`
> The drive is initializing or recovering from errors. Wait for the drive to go online.

vol_id
> is the identifier of the cartridge in the transport. The cartridge id appears only if the *drive_stat* is In use.

drive_stat
> is one of the following drive statuses:

`In use`
> The drive has a cartridge mounted or is reserved for a mount.

`Available`
> The drive is available for a mount.

drive_type
> is the drive type.

Examples

To display status of compatible transports listed by proximity to the largest concentration of scratch tapes in pool 5:

```
query mount * 5
```

To display status of compatible transports listed by proximity to the largest concentration of 3480 scratch tapes in common pool 0:

```
query mount * 0 media 3480
```

See Also

For information about...	See...
Creating or modifying scratch pools	"define pool" on page 200
Deleting empty scratch pools	"delete pool" on page 202
Mounting a scratch cartridge on a transport	"mount *" on page 223
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Displaying scratch pool attributes	"query pool" on page 250
Displaying scratch cartridge status	"query scratch" on page 254
Setting or clearing cartridge scratch attributes	"set scratch" on page 267
Changing the state of a library component	"vary" on page 277

■ query pool

The `query pool` command displays scratch pool attributes.

Format

`query pool` *pool_id*`... | all`

Options

pool_id `| all`
    specifies the scratch pool to query or all for all pools. Pool 0 is the common pool.

Usage

Use the `query pool` command to display scratch pool attributes in the following format:

```
yyy-mm-dd hh:mm:ss Pool Status
Identifier Volume Count Low Water Mark High Water Mark  A ttributes

pool_id     vol_count     low_water_mark high_water_mark attribute
```

Where:

pool_id

is the specified scratch pool.

vol_count
>   is the number of scratch cartridges in the pool.
>
>   Absent and ejected scratch cartridges are not counted.

low_water_mark
>   is the low cartridge warning threshold. If the scratch cartridge count falls below this threshold, ACSLS logs a warning message in the event log. A "-" after the value means that the scratch cartridge count is below the low cartridge threshold.

high_water_mark
>   is the high cartridge warning threshold. If the scratch cartridge count reaches or exceeds this threshold, ACSLS logs a warning message in the event log. A "+" after the value means that the scratch cartridge count is at or above the high cartridge threshold.

attribute
>   is displayed if `overflow` is set (via the `set scratch` command) for the specified scratch pool. overflow specifies that scratch cartridges are selected from the common scratch pool (Pool 0) if `mount scratch *` requests cannot be satisfied with a cartridge from the specified scratch pool.

Examples

To display status information for scratch pool 5:

query pool 5

To display status information for all scratch pools:

```
query pool all
```

See Also

For information about...	See...
Creating or modifying scratch pools	"define pool" on page 200
Deleting empty scratch pools	"delete pool" on page 202
Displaying scratch cartridge status	"query scratch" on page 254
Setting or clearing cartridge scratch attributes	"set scratch" on page 267

■ query port

The `query port` command displays port status.

Format

```
query port port_id... | all
```

Options

port_id | all
    specifies the port to query or all for all ports.

Usage

Use the query port command to display the status of a port in the following format:

```
yyy-mm-dd  hh:mm:ss  Port Status
State    Identifier

state    port_id
```

Where:

state
    is one of the following port states:

    online
        The port is online.

    offline
        The port is offline.

port_id
    is the port identifier.

Examples

To display status information for port 0,0:

```
query port 0,0
```

To query all ports:

```
query port all
```

■ query request

The `query request` command displays request status.

Format

query request *request_id*... | all

Options

request_id | all
   specifies the request to query or `all` for all requests.

Usage

Use the `query request` command to display the status of a request in the following format:

```
yyy-mm-dd  hh:mm:ss  Request Status
Identifier    Command    Status

request_id    command    status
```

Where:

request_id
   is the ACSLS request identifier.

command
   is the ACSLS command that corresponds to the request identifier.

status
   is one of the following request statuses:

   Current
      ACSLS is processing the request.

   Pending
      The request is waiting to be processed.

   Not found

The specified request is not a valid ACSLS request.

Examples

To display status information for request 33179:

```
query request 33179
```

To display all current and pending requests:

query request all

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194

■ query scratch

The query scratch command displays the status of scratch cartridges in a pool sorted by ascending order of access date. The cartridges with the earliest access date are at the top of the list with the most recently used cartridges at the bottom of the list. Pool IDs in the first column do not appear in any particular order. Only those cartridges permitted through Access Control are displayed.

Format

query scratch *pool_id*... | all

Options

pool_id | al̲l

specifies the scratch pool to query or all for all pools. Pool 0 is the common pool.

Usage

Use the query scratch command to display the status of scratch cartridges in a pool in the following format:

```
yyy-mm-dd  hh:mm:ssScratch Status
Scratch Pool    Identifier     Home Location    Status    Type
pool_id         vol_id         cell_id          status    media_type
```

Where:

pool_id

is the specified scratch pool.

vol_id
is the identifier of the scratch cartridge.

cell_id
is the storage cell that contains the cartridge.

status
is the location of the cartridge:

`home`
The cartridge is in a storage cell.

`in drive`
The cartridge is in a transport.

`in transit`
The cartridge is being moved.

media_type
is the cartridge's media type (for example, 3480, 3490E, DD3D, or DLTIV).

Note: Scratch cartridges with a cartridge status of absent, ejected, or missing are not included in the `query scratch` output.

To see cartridges sorted by pool, you issue successive Query Scratch commands for each pool. Alternately, you can issue the Display volume command to see the information and sort it by pool, but you will not be guaranteed of getting the least recently used scratch cartridge.

Examples

To display status information for scratch cartridges in scratch pool 29015:

`query scratch 29015`

To display status information for scratch cartridges in all scratch pools:

query scratch all

See Also

For information about...	See...
Creating or modifying scratch pools	"define pool" on page 200
Deleting empty scratch pools	"delete pool" on page 202
Setting or clearing cartridge scratch attributes	"set scratch" on page 267

■ query server

The `query server` command displays ACSLS and library status.

Format

`query server`

Options

None.

Usage

Use the `query server` command to display the status of ACSLS and the library in the following format:

```
yyy-mm-ddhh:mm:ssServer Status
IdentifierStateFree CellAudit   Mount    Dismount Enter    Eject
           Count    C/P         C/P      C/P      C/P      C/P

    state  count    n/n         n/n      n/n      n/n      n/n
```

Where:

`Identifier`
    is blank

`state`
    is one of the following ACSLS states:

`idle`
    ACSLS is idle (not processing requests).

`idle pending`
    ACSLS will process current and pending request, reject new request, then go idle.

`recovery`

> ACSLS is initializing (going to run state) or recovering from errors; ACSLS is not processing requests.

`run`
> ACSLS is running (processing requests).

count
> is the number of free storage cells in the library.

n

> is the number of current (C) and pending (P) ACSLS requests for each command that requires library resources (`audit`, `mount`, `dismount`, `enter`, and `eject`).

Hint: You cannot cancel a `query server` request.

Examples

To display status information on the server:

`query server`

See Also

For information about...	See...
Changing the state of a library component	"vary" on page 277
Displaying request status	"query request" on page 253

■ query volume

The `query volume` command displays the location of a cartridge.

Format

`query volume` *vol_id*`... |` `all`

Options

vol_id `|` `all`
> specifies the volume to query or `all` for all volumes.

Usage

Use the `query volume` command to display the location of a volume in the following format:

```
yyy-mm-dd  hh:mm:ss  Volume Status
Identifier  Status  Current Location  Type

vol_id     status     location      media_type
```

Where:

`vol_id`
is the volume identifier.

`status`
is the location of the cartridge.

`home`
The cartridge is in a storage cell.

`in drive`
The cartridge is in a transport.

`in transit`
The cartridge is being moved or is missing.

`absent`
The cartridge cannot be found.

`ejected`
The cartridge has been ejected from the library.

`location`
specifies the location as one of the following:

- If the status is home, the location is a storage cell identifier.

- If the status is `in transit`, the location is either a cell identifier or a drive identifier.

- If the status is `in drive`, the location is a drive identifier.

`media_type`
is the volume's media type (for example, 3480, 3490E, DD3D, or DLTIV).

Examples

To display status information for volume 2903B:

query volume 2903B

To display all volumes in the library:

    q volume all

See Also

For information about...	See...
Other volume information	"Creating a Custom Volume Report" on page 121
Display Command	"display Commands" on page 288 and "display volume" on page 304
Mounting a data volume on a transport	"mount" on page 220
Displaying the lock status of a drive or volume	"query lock" on page 241
Displaying the status of media-compatible transports for a specified data volume	"query mount" on page 245
Displaying scratch pool attributes	"query pool" on page 250
Setting or clearing volume scratch attributes	"set scratch" on page 267
Changing the state of a library component	"vary" on page 277

■ set commands

The <u>set</u> commands set various attributes of different library components. See the following sections for more information about each <u>set</u> command, including format, options, usage, and messages.

Format

The following shows the general format of the set commands:

<u>set</u> type [<u>of</u>f | *subtype*] [*] *identifier*...

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194

Command Area Messages

Success Messages

The following message appears when a set command succeeds:

```
Set: Set completed, Success.
```

See each <u>set</u> command for specific success messages.

Intermediate Messages

None.

Error Messages

See each <u>set</u> command for specific messages.

Display Area Messages

See each <u>set</u> command for specific messages.

■ set cap mode

The <u>set</u> <u>cap</u> <u>mod</u>e command sets a CAP's entry mode.

Format

<u>set</u> <u>cap</u> <u>mod</u>e *cap_mode cap_id*

Options

cap_mode
   specifies manual or automatic CAP mode.

cap_id
   specifies the CAP identifier.

   You cannot specify a CAP identifier that contains an asterisk (*) unless the priority is set.

Usage

Use the <u>set</u> <u>cap</u> <u>mod</u>e command to set a CAP's entry mode to one of the following modes:

manual
   You must enter an enter command before entering cartridges.

automatic
   You can enter cartridges without first entering an enter command

   Note: You cannot set the CAP mode to automatic in a partitioned library.

Hint: You cannot change a CAP's mode while the CAP is in use. That is, if the door is open during either manual or automatic enter operations, you cannot change its mode until you complete the enter operation.

Examples

To set CAP 0,3,1 in manual mode:

```
set cap mode manual 0,3,1
```

To set CAP 0,3,1 in automatic mode:

```
set cap mode automatic 0,3,1
```

See Also

For information about...	See...
Making a CAP (manual mode) ready to enter labelled cartridges into the library	"enter" on page 211
Displaying CAP status	"query cap" on page 233
Changing the state of a library component	"vary" on page 277
Making a CAP ready to enter unlabeled cartridges into the library	"venter" on page 284

Command Area Messages

Success Messages

Set: CAP *cap_id*, mode changed to *cap_mode*.

> **Explanation:** ACSLS changed the mode of the specified CAP.

> **Variable:**

> - *cap_id* is the CAP whose mode is changed.

> - *cap_mode* is the new enter mode of the CAP.

Intermediate Messages

None.

Error Messages

Set: CAP *cap_id* Set failed, Incorrect attribute.

> **Explanation:** ACSLS cannot change the mode of the specified CAP because you specified an invalid CAP mode.

> **Variable:** *cap_id* is the CAP whose mode is not changed.

CAP *cap_id*: Automatic mode.

> **Explanation:** ACSLS changed the specified CAP's mode to automatic.

> **Variable:** *cap_id* is the CAP whose mode is changed.

CAP *cap_id*: Manual mode.

> **Explanation:** ACSLS changed the specified CAP's mode to manual.

Variable: `cap_id` is the CAP whose mode is changed.

■ set cap priority

The set cap priority command sets a CAP's automatic selection priority.

Format

set cap priority *cap_priority cap_id*

Options

cap_priority
   specifies the CAP priority. Valid values are 0 to 16, where 16 is the highest priority. All CAPs initially have a 0 priority, which means that ACSLS does not automatically select the CAP.

cap_id
   specifies the CAP identifier. You must specify a particular CAP; you cannot specify an asterisk (*) to set the same priority for all CAPs.

Usage

Use the set cap priority command to set a CAP's automatic selection priority.

If a CAP request specifies an asterisk (*) for the CAP ID, ACSLS automatically selects an available CAP with highest non-zero priority for each ACS specified in the request.

Examples

To assign priority 16 to CAP 0,3,1:

   set cap priority 16 0,3,1

See Also

For information about...	See...
Updating the ACSLS database to match the actual inventory of library cartridges	"audit" on page 188
Ejecting cartridges from the library	"eject" on page 207
Making a CAP (manual mode) ready to enter labelled cartridges into the library	"enter" on page 211
Displaying CAP status	"query cap" on page 233
Changing the state of a library component	"vary" on page 277
Making a CAP ready to enter unlabeled cartridges into the library	"venter" on page 284

Command Area Messages

Updates the ACSLS database to match the actual inventory of library cartridges.

Success Messages

`Set: CAP cap_id, priority changed to cap_priority.`

> **Explanation:** ACSLS changed the priority of the specified CAP.

> **Variable:**

> - `cap_id` is the CAP whose priority is changed.

> - `cap_priority` is the new CAP priority.

Intermediate Messages

None.

Error Messages

`Set: CAP cap_id Set failed, Incorrect attribute.`

> **Explanation:** ACSLS cannot change the mode of the specified CAP because you specified an invalid CAP priority.

> **Variable:** `cap_id` is the CAP whose priority is not changed.

Display Area Messages

None.

■ set clean

The <u>set</u> <u>clean</u> command sets cleaning cartridge attributes.

Format

<u>set</u> <u>clean</u> *max_usage* | <u>off</u> *vol_id* | *volrange*

Options

max_usage | <u>off</u>
specifies the number of times a cleaning cartridge is used before ACSLS stops selecting the cartridge to clean transports. off specifies that ACSLS will not select the cartridge and redefines the cartridge as a data cartridge.

> **Note:** You cannot:
>
> • specify a maximum cleaning usage for a media type that is reserved for only data cartridges
>
> • set cleaning *off* for a media type that is only a cleaning cartridge

vol_id | *volrange*
specifies the cleaning cartridge or range of cartridges.

Usage

Use the <u>set</u> <u>clean</u> command to set the number of times ACSLS will select a cleaning cartridge. You also use set clean to set a cartridge's cleaning cartridge attribute off, which you do if you have incorrectly defined a data cartridge as a cleaning cartridge.

Examples

To set the maximum use to 10 for cleaning cartridges CLN108 - CLN112:

set clean 10 CLN108-CLN112

To set the cleaning cartridge attribute off and redefine cartridges HRR234 - HRR244 as data cartridges:

```
set clean off HRR234-HRR244
```

See Also

For information about...	See...
Guidelines and procedures for cleaning transports	"Cleaning Cartridges" on page 97
Displaying cleaning cartridge status	"query clean" on page 235

Command Area Messages

Success Messages

Set: volume *vol_id* is a cleaning cartridge.

Explanation: ACSLS specified that the cleaning cartridge is valid for drive cleaning.

Variable: *vol_id* is the valid cleaning cartridge.

Set: volume *vol_id* is not a cleaning cartridge.

Explanation: ACSLS specified that the cleaning cartridge is not valid for drive cleaning.

Variable: *vol_id* is the invalid cleaning cartridge.

Intermediate Messages

None.

Error Messages

Set: Clean vol_id Set failed, Incorrect attribute.

Explanation: ACSLS cannot change the cleaning attribute because the specified cartridge is not a cleaning cartridge.

Variable: *vol_id* is the data or scratch cartridge.

Display Area Messages

None.

■ set lock

The <u>set loc</u>k command sets your lock ID.

Format

<u>set loc</u>k *lock_id*

Options

lock_id
specifies the lock ID. Valid lock IDs are 0 to 32767.

Usage

Use the set lock command to set or change your lock ID. Change your current lock ID when you want to remove locks on a drive or cartridge whose lock ID does not match your current lock ID as follows:

- To clear all locks on a specified drive or cartridge, set your lock ID to 0, then enter a <u>clear loc</u>k command.

- To remove active locks on transports or cartridges, set your lock ID to the lock ID of the locked components, then enter an unlock command.

Note: When you enter a <u>loc</u>k command to lock a cartridge or transport, ACSLS assigns a lock ID to the cartridge or drive and then changes your lock ID to the lock ID of the cartridge or transport. You cannot use the <u>set loc</u>k command to set your lock ID and then use the <u>loc</u>k command to lock a cartridge or drive with the lock ID that you set with <u>set loc</u>k.

Examples

To change your current lock ID lock to new lock ID 354:

```
set lock 354
```

See Also

For information about...	See...
Removing all active or pending locks on a specified drive or cartridge	"clear lock" on page 198
Locking transports and cartridges	"lock" on page 217
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Displaying your lock or user ID	"show" on page 270
Removing active locks	"unlock" on page 275

Command Area Messages

Success Messages

```
Set: Changed lock identifier from previous_lock_id to
lock_id.
```

Explanation: ACSLS changed your lock ID.

Variable:

previous_lock_id was your previous lock ID.

lock_id is your new lock ID.

Intermediate Messages

None.

Error Messages

None.

Display Area Messages

None.

■ set owner

The <u>set</u> <u>owner</u> command sets volume ownership.

Format

<u>set</u> <u>owner</u> *owner_id* volume *vol_id* | *volrange*

Options

owner_id
   specifies the owner identifier. You must enclose this value in quotes ("").

<u>vol</u>ume *vol_id* | *volrange*
   specifies the volume or range of volumes.

Usage

Use the <u>set</u> <u>owner</u> command to set volume ownership. You must enter set owner from cmd_proc; you cannot set volume ownership from a client application.

Examples

To assign ownership of volume YUMA06:

   set owner "cray" volume YUMA06

Note:

   To remove volumn ownership of volume YUMA06:

   set owner "" volume 0YUMA06

Command Area Messages

None.

Display Area Messages

None.

■ set scratch

The <u>set</u> <u>scr</u>atch command sets or clears a volume's scratch attribute and assigns the volume to a pool.

Format

<u>set</u> <u>sc</u>ratch [<u>off</u>] *pool_id vol_id* | *volrange*

Options

off
> specifies that the volume is a data cartridge.

pool_id
> specifies the cartridge's scratch pool. Specify an asterisk (*) to reassign a data cartridge to its current pool.

vol_id | volrange
> specifies the volume or range of volumes.

Usage

Use the set scratch command to set or clear a volume's scratch attribute and assign the volume to a pool.

Examples

To define volumes YUMA10-YUMA20 as scratch volumes and assign them to scratch pool 5:

```
set scratch 5 YUMA10-YUMA20
```

To move scratch volumes YUMA10-YUMA15 to pool 10:

```
set scratch 10 YUMA10-YUMA15
```

To "unscratch" (change from scratch to data) volumes YUMA16-YUMA20 and move them to the common pool (pool 0):

```
set scratch off 0 YUMA16-YUMA20
```

To scratch to data volumes YUMA16-YUMA20 and keep them in their current pool:

```
set scratch * YUMA16-YUMA20
```

See Also

For information about...	See...
Displaying scratch cartridge status	"query scratch" on page 254
Displaying scratch pool attributes	"query pool" on page 250
Creating or modifying scratch pools	"define pool" on page 200
Deleting empty scratch pools	"delete pool" on page 202

Command Area Messages

Success Messages

`Set: volume` *`vol_id`* `in tape pool` *`pool_id`* `is a scratch cartridge.`

> **Explanation:** ACSLS changed the specified data cartridge to a scratch cartridge and assigned it to a pool.

> **Variable:**

> - *`vol_id`* is the specified volume.

> - *`pool_id`* is the pool to which the volume is assigned.

`Set: volume` *`vol_id`* `in tape pool` *`pool_id`* `is a data volume.`

> **Explanation:** ACSLS changed the specified scratch cartridge to a data cartridge.

> **Variable:** *`vol_id`* is the specified volume.

Intermediate Messages

None.

Error Messages

`Set: Scratch vol_id Set failed, Incorrect attribute.`

> **Explanation:** ACSLS cannot change the scratch attribute because the specified cartridge is a cleaning cartridge.

> **Variable:** *`vol_id`* is the cleaning cartridge.

Display Area Messages

`Pool` *`pool_id`*`: low water mark warning.`

> **Explanation:** The number of cartridges in the specified scratch pool is less than or equal to the low threshold.

> **Variable:** *`low_water_mark`* is the low threshold of the specified scratch pool.

`Pool` *`pool_id`*`: high water mark warning.`

> **Explanation:** The number of cartridges in the specified scratch pool is greater than or equal to the high threshold.

> **Variable:** *`high_water_mark`* is the high threshold of the specified scratch pool.

■ **show**

The <u>sh</u>ow command displays your lock ID or user ID.

Format

<u>sh</u>ow *type*

Options

type
    specifies one of the following types:

    <u>loc</u>k
        Your lock ID.

    <u>us</u>er
        Your user ID.

Usage

Use the <u>sh</u>ow command to display your lock ID or user ID.

Examples

To display requestor's user_id:

show user

To display current lock_id:

show lock

Notes

None.

See Also

For information about...	See...
Removing all active or pending locks on a specified drive or cartridge	"clear lock" on page 198
Locking transports and cartridges	"lock" on page 217
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Setting your lock ID	"set lock" on page 265
Removing active locks	"unlock" on page 275

Command Area Messages

Success Messages

Show: Current lock identifier is *lock_id.*

>**Explanation:** Your current lock ID is lock_id.

Show: User identifier is user_id.

>**Explanation:** Your current user ID is user_id.

Intermediate Messages

None.

Error Messages

None.

Display Area Messages

None.

■ start

The start command starts ACSLS request processing.

Format

start

Options

None.

Usage

Use the start command to put ACSLS in run state and start ACSLS request processing. You typically use the start command to restart request processing if ACSLS is idled.

Examples

To restart ACSLS request processing:

>**start**

If ACSLS is in run state, entering the start command has no effect.

See Also

For information about...	See...
Stopping ACSLS from processing new requests	"idle" on page 215
Displaying the status of a library component	"query commands" on page 230
Restarting ACSLS	"Starting ACSLS" on page 2
Stopping ACSLS	"Stopping ACSLS" on page 3
Changing the state of a library component	"vary" on page 277

Command Area Messages

Success Messages

```
ACSLM Request Processing Started: Success.
```

Explanation: ACSLS started processing requests.

Intermediate Messages

None.

Error Messages

```
ACSLM Request Processing Not Started: status
```

Explanation: ACSLS did not start processing requests.

Variable: *status* is the reason for the failure. For more information on common status messages, see ACSLS Messages.

Display Area Messages

```
Server system running
```

Explanation: ACSLS started processing requests.

■ switch lmu

In dual-LMU configurations, the `switch lmu` command manually switches ACS management from the ACS's master LMU to the standby LMU.

Note: ACSLS supports dual-LMU configurations for only the 9330 LMU with host/LMU microcode compatibility Level 11 or above loaded. The same microcode level must be loaded in both LMUs.

Format

```
switch lmu acs_id
```

Options

acs_id

>   specifies the ACS to switch ACS management from the master to the standby LMU.

Usage

Use the switch lmu command to manually switch ACS management from the ACS's master LMU to the standby LMU. Before you enter a switch lmu command, make sure that:

*   ACSLS is in run state

*   The ACS you specify is in online or diagnostic state

*   At least one port is online to each LMU

Examples

In dual-LMU configurations, you can use the switch lmu command to switch ACS management from the ACS's master LMU to the standby LMU. Assume the following dual-LMU configuration:

*   LMU A is in the master role and LMU B is in the standby role.

*   Both LMUs manage ACS 0.

*   The ports to LMU A have port IDs 0,0 and 0,1.

The following procedure tells how to use the switch lmu command to let you replace a LAN cable between the LMUs.

To replace a LAN cable between dual LMUs configured as described above, do the following:

1. Switch to LMU B:

    switch lmu 0

2. Vary the ports to LMU A offline:

    vary port 0,0 0,1 offline

3. Replace the LAN cable.

4. Vary the ports to LMU A back online:

    vary port 0,0 0,1 online

5. Switch back to LMU A:

    switch lmu 0

See Also

For information about...	See...
Displaying ACS status	"query acs" on page 231
Displaying LMU status	"query lmu" on page 238
Displaying port status	"query port" on page 251
Displaying ACSLS and library status	"query server" on page 256
Starting ACSLS request processing	"start" on page 271
Restarting ACSLS	"Starting ACSLS" on page 2
Changing the state of a library component	"vary" on page 277

Command Area Messages

Success Messages

`Switch: Switch lmu completed for` *acs_id*`, Success.`

> **Explanation:** ACSLS switched ACS management from the master to the standby LMU.

> **Variable:** *acs_id* is the ACS whose LMUs switched roles.

`Switch: Switch lmu initiated for` *acs_id*`, Success.`

> **Explanation:** ACSLS started switching ACS management from the master to the standby LMU.

> **Variable:** *acs_id* is the ACS whose LMUs are switching roles.

`Switch: Switch lmu failed,` *acs_id* `not found.`

> **Explanation:** You specified an invalid ACS.

> **Variable:** *acs_id* is the ACS identifier you specified on the <u>sw</u>itch <u>lm</u>u command.

`Switch: Switch lmu failed,` *acs_id* `is offline.`

> **Explanation:** You specified an ACS that is offline. Use the vary command to vary the ACS to online or diagnostic mode.

> **Variable:** *acs_id* is the ACS identifier you specified on the <u>sw</u>itch <u>lm</u>u command.

```
Switch: Switch lmu failed, acs_id not configured for dual
lmu.
```

> **Explanation:** You specified an ACS that is not configured with master and standby LMUs. Either:
>
> • Reconfigure the specified ACS, then reenter the <u>sw</u>itch <u>lm</u>u command
>
> • Enter a <u>sw</u>itch <u>lm</u>u command that specifies an LMU that is configured with master and standby LMUs.
>
> **Variable:** *acs_id* is the ACS identifier you specified on the <u>sw</u>itch <u>lm</u>u command.

```
Switch: Switch lmu failed, not communicating.
```

> **Explanation:** An LMU switch failed because the standby LMU is not communicating. Check the standby LMU for communications problems.

```
Switch: Switch lmu failed, switch already active.
```

> **Explanation:** An LMU switch is in progress for the ACS you specified. Wait until the switch completes, then, if desired, reenter the <u>sw</u>itch <u>lm</u>u command.

```
Switch: Switch lmu failed, port is offline.
```

> **Explanation:** You specified an ACS with no LMU ports online Vary at least one port online to each LMU.

Display Area Messages

None.

■ **unlock**

The <u>un</u>lock command removes active locks (associated with your current lock ID) on a specified drive or cartridge or removes all active locks.

Format

<u>un</u>lock *type identifier*...| <u>al</u>l

Options

type identifier

specifies a library component. Table 21 lists the components that you can unlock.

Table 21. Valid Components for Unlock

Library Component	type	identifier
transport	<u>dr</u>ive	*drive_id*
volume	<u>vol</u>ume	*vol_id*

<u>al</u>l
    specifies all active locks.

Usage

Use the <u>un</u>lock command to remove active locks on specified cartridges and transports or remove all active locks. The lock ID of the component must match your current lock ID.

When you unlock a component that is not in use, ACSLS checks for pending locks on the component. If there is a pending lock, ACSLS locks the component with the lock ID of the pending lock.

Note: The <u>un</u>lock command always resets your lock ID to 0.

Examples

To remove active locks on drive 0,0,2,0:

    unlock drive 0,0,2,0

To remove active locks on all locked cartridges:

    unlock volume all

See Also

For information about...	See...
Removing all active or pending locks on a specified drive or volume	"clear lock" on page 198
Locking transports and volumes	"lock" on page 217
Displaying the lock status of a drive or cartridge	"query lock" on page 241
Setting your lock ID	"set lock" on page 265
Displaying your lock or user ID	"show" on page 270

Command Area Messages

Success Messages

Unlock: Unlock completed, Success.

```
Unlock: drive drive_id unlocked.
```

Explanation: ACSLS unlocked the specified transport.

Variable: *drive_id* is the unlocked transport.

```
Unlock: Unlock completed, Success.
   Unlock: volume vol_id unlocked.
```

Explanation: ACSLS unlocked the specified volume.

Variable: *vol_id* is the unlocked volume.

Intermediate Messages

None.

Error Messages

```
Unlock: Unlock of drive drive_id failed, status.
```

Explanation: ACSLS cannot unlock the transport.

Variable: status is the reason for the failure. For more information on common status messages, see ACSLS Messages.

```
Unlock: Unlock of volume vol_id failed, status.
```

Explanation: ACSLS cannot unlock the volume.

Variable: *status* is the reason for the failure. For more information on common status messages, see ACSLS Messages.

Display Area Messages

None.

■ vary

The <u>va</u>ry command changes the state of a library component.

Format

<u>va</u>ry *type identifier*...*state* [<u>f</u>orce]

Options

type [*identifier*]
   specifies a library component. Table 22 lists the components you can vary.

Table 22. Valid Components for Vary

ACS Component	type	identifier
ACS	<u>ac</u>s	*acs_id*

Table 22. Valid Components for Vary

ACS Component	type	identifier
LSM	<u>ls</u>m	*acs_id,lsm_id*
CAP	<u>ca</u>p	*cap_id*
transport	<u>dr</u>ive	*drive_id*
port	<u>por</u>t	*port_id*

state
> specifies one of the following device states: <u>on</u>line (online to clients and cmd_proc), <u>of</u>fline, or <u>dia</u>gnostic (online to cmd_proc only).
>
> You can vary an ACS, LSM, CAP, or drive online, offline, or diagnostic. You can vary a port either online or offline. See "Usage" for more information about device states.

force
> varies an ACS, LSM, CAP, or drive <u>of</u>fline after processing only the current robotic request.

Usage

Use the <u>va</u>ry command to change the state of an ACS, LSM, CAP, transport, or port. The following sections tell how each of the device states you can specify affects library components.

vary offline

Table 23 shows the results of <u>va</u>ry <u>of</u>fline for each ACS component. When you vary a component offline and it first goes to offline-pending state, ACSLS processes all active and pending requests for the component, rejects new requests, then varies the component offline.

Table 23. Vary Offline Results

ACS Component	Results
ACS	ACS and subcomponents go offline-pending, then offline. LSMs must go offline before the ACS goes offline.
LSM	LSM goes offline-pending, then offline.
CAP	CAP goes offline-pending, then offline.
transport	If available, drive goes offline immediately. If in use, drive stays online.

Table 23. Vary Offline Results

ACS Component	Results
port	Port goes offline immediately if:
	• ACS has other ports online, or
	• ACS is offline.
	Otherwise, the port stays online.

vary offline force

Table 24 shows the results of `vary offline force` for each ACS component. This option is useful if the device must be taken to the offline state while it is involved in extended activity (for example, during an audit).

Table 24. Vary offline force Results

ACS Component	Results
ACS	Only the current robotic request completes, then the ACS and subcomponents go offline immediately. Pending requests are discarded and new requests are rejected. LSMs must go offline before the ACS goes offline.
LSM	Only the current robotic request completes, then the LSM goes offline immediately. Pending requests are discarded and new requests are rejected.
CAP	Only the current robotic request completes, then the CAP goes offline immediately. Pending requests are discarded and new requests are rejected.
transport	Only the current robotic request completes, then the drive goes offline immediately. Pending requests are discarded and new requests are rejected.
port	Not valid.

vary diagnostic

The `vary diagnostic` request places the specified component in the diagnostic state (online to `cmd_proc` only). ACSLS processes all active and pending requests for the component, rejects new client application requests, then varies the component to diagnostic state. For an ACS, all of its subordinate LSMs are also varied to diagnostic.

vary online

Table 25 shows the results of `vary online` for each ACS component. When you vary a component online and it first goes to recovery state, ACSLS processes all active and pending requests for the component, rejects new

requests, then varies the component online. When the component goes to online, ACSLS processes all requests for the component.

Table 25. Vary Online Results

ACS Component	Results
ACS	If ACS is offline, ACS and its LSM go to recovery, then online.
	If ACS is in diagnostic state, ACS and its LSM go to online immediately.
LSM	LSM goes to recovery, then online. You cannot vary an LSM online if it is attached to an offline ACS.
CAP	CAP goes to recovery, then online.
transport	drive goes to recovery, then online.
port	Port goes online immediately.

Examples

To vary drive 0,0,9,3 offline:

```
vary drive 0,0,9,3 offline
```

To vary CAP 0,0,0 diagnostic:

```
vary cap 0,0,0 diagnostic
```

To force lsm 0,1 offline:

```
vary lsm 0,1 offline force
```

Notes

- IPLing the system does not change the state of these components. Installing or reconfiguring ACSLS places all components in the online state wherever possible.

- When two or more partitions of the same library are managed by ACSLS, use separate vary commands to vary the same CAP (identified as being in differenct ACSs).

    Example: If ACS 0 and ACS 1 are two partitions of the same library, vary the CAP online or offline independently for each partition (and each ACS). The same CAP can be varied online to one partition (ACS) and offline to another partition (ACS).

See Also

For information about...	See...
Displaying the status of a library component	"query commands" on page 230

Command Area Messages

Success Messages

`Vary:` *`type identifier`* `Varied` *`state`*

Explanation: ACSLS changed the state of the specified library component.

Variable:

- *`type`* is the library component type.

- *`identifier`* is the library component identifier.

- *`state`* is the new stat of the library component.

Intermediate Messages

None.

Error Messages

`In-transit volume recovery incomplete.`

Explanation: The LSM robot cannot dispose of in-transit volumes while the LSM or its ACS is varied online. If you are varying an ACS online, the ACS state is changed to online immediately, but any LSMs that are unable to complete in-transit volume recovery remain offline.

`Vary: Vary type identifier failed, Drive in use.`

Explanation: ACSLS cannot change the state of the specified library component because the command involved an in-use transport.

Variable:

- *`type`* is the library component type.

- *`identifier`* is the library component identifier.

`Vary: Vary type identifier failed, Incorrect lockid`

Explanation: ACSLS cannot change the state of the specified library component because the drive lock_id does not match your lock_id.

Variable:

- type is the library component type.

- identifier is the library component identifier.

Vary: `Vary type identifier failed, State unchanged.`

Explanation: ACSLS cannot change the state of the specified library component because the component is already in the requested state.

Variable:

- *type* is the library component type.

- *identifier* is the library component identifier.

Vary: `Vary type identifier failed, Vary disallowed.`

Explanation: ACSLS cannot change the state of the specified library component because a vary diagnostic request was issued from a client application.

Variable:

- *type* is the library component type.

- *identifier* is the library component identifier.

Vary: `Vary type identifier failed, Incorrect state.`

Explanation: ACSLS cannot change the state of the specified library component because you tried to vary:

- A CAP online when the LSM or ACS is in the diagnostic or offline state.

- A CAP diagnostic when the LSM or ACS is offline.

- An LSM online when the ACS is in the diagnostic or offline state.

- An LSM diagnostic if the ACS is offline.

Variable:

- type is the library component type.

- identifier is the library component identifier.

Vary: `Vary type identifier failed, Vary in progress.`

Explanation: ACSLS cannot change the state of the specified library component because

- A CAP and the CAP or its LSM or ACS is in a transitory (recovery or offline-pending) state.

- An LSM and the LSM, its ACS, or a subordinate CAP is in a transitory state.

- An ACS and the ACS, a subordinate LSM, or a subordinate CAP is in a transitory state.

- An ACS, LSM, or CAP to the diagnostic or online state and a vary offline/force request overrode the attempt.

- A drive which is in a transition state of recovery.

Variable:

- *type* is the library component type.

- *identifier* is the library component identifier.

`Vary: Vary type identifier failed, State unchanged.`

Explanation: ACSLS cannot change the state of the specified library component because you tried to vary an ACS, LSM, or CAP offline and a vary offline/force request overrode the attempt.

Variable:

- *type* is the library component type.

- *identifier* is the library component identifier.

`Vary: Vary type identifier failed, ACS online.`

Explanation: ACSLS cannot change the state of the specified library component because you tried to vary the last port offline in an online ACS.

Variable:

- type is the library component type.

- identifier is the library component identifier.

`Vary: Vary type identifier failed, No ports online.`

Explanation: ACSLS cannot change the state of the specified library component because there are no ports online for an ACS.

Variable:

- *type* is the library component type.

- *identifier* is the library component identifier.

Display Area Messages

One of the following messages appears when an ACS, LSM, transport, or port changes state.

```
ACS acs_id: state
LSM lsm_id: state
Drive drive_id: state
Port port_id: state
```

The following message appears if, during recovery of in-transit volumes, a cartridge label is unreadable or duplicate. The cartridge is moved to the CAP.

CAP *cap_id*. Remove volumes from CAP.

cap_id
    is the CAP containing the cartridges.

■ venter

The <u>ve</u>nter command makes a CAP ready to enter unlabeled cartridges into the library.

Note: The L5500, SL500, and SL8500 libraries do not support unlabeled cartridges (venters).

Format

<u>ve</u>nter *cap_id vol_id*

Options

cap_id
    specifies the CAP through which the cartridges are to be entered.

vol_id
    specifies the virtual labels of the cartridges to be entered.

Usage

Use the <u>ve</u>nter command to make a CAP ready to enter cartridges with missing or unreadable labels into the library. You specify virtual labels (volume IDs) on the venter command, which adds cartridge information to the ACSLS database. From one to 42 cartridges can be entered at a time.

Unless the media type is "3480", you cannot venter a cartridge that has no separate media type label.

CAUTION: ACSLS only supports the venter command for SCSI-attached LSMs when both:

- the cartridge being ventered has a media label, and

- the cartridge's media label is compatible with drives attached to the SCSI library.

Unless the cartridge's media type is 3480, you cannot venter a cartridge without a separate media label.

Use the <u>ve</u>nter command to enter cartridges with missing or unreadable labels. Do *not* open the LSM door and place cartridges with missing or unreadable labels in a storage cell because ACSLS cannot manage these cartridges. During an

audit, ACSLS will eject cartridges with missing or unreadable labels and no virtual labels.

The <u>v</u>enter command does not provide an option to specify the media type of the cartridge you want to enter. In a mixed-media environment, ACSLS cannot prevent transport/media incompatibilities for virtually entered cartridges!

Examples

To assign virtual labels to volumes MAINT1 and MAINT2 and enter these volumes through CAP 0,2,2:

```
venter 0,2,2 MAINT1 MAINT2
```

See Also

For information about...	See...
Cancelling a command	"cancel" on page 194
Creating or modifying scratch pools	"define pool" on page 200
Ejecting cartridges from the library	"eject" on page 207
Making a CAP (manual mode) ready to enter labelled cartridges into the library	"enter" on page 211
Mounting a scratch cartridge on a transport	"mount *" on page 223
Mounting a data cartridge on a transport	"mount" on page 220
Displaying CAP status	"query cap" on page 233
Setting CAP mode (manual or automatic)	"set cap mode" on page 260
Setting CAP selection priority	"set cap priority" on page 262

Command Area Messages

Success Messages

Venter: Enter complete, *nn* cartridges entered.

> **Explanation:** ACSLS entered the displayed number of cartridges.

> **Variable:** *nn* is the total number of cartridges entered.

Venter: *vol_id* Entered through *cap_id*

> **Explanation:** ACSLS entered the specified virtual cartridge.

> **Variable:**

- *vol_id* is the virtual cartridge label.

- *cap_id* is the CAP through which the cartridge was entered.

Intermediate Messages

None.

Error Messages

Venter: *vol_id* Enter failed, *status*

Explanation: ACSLS did not enter the specified virtual cartridge.

Variable:

- *vol_id* is the virtual cartridge label.

- The *status* is the translated completion status of the request returned by ACSLS.

Display Area Messages

CAP *cap_id*: Place cartridges in CAP.

Explanation: The CAP is ready for cartridge entry. Open the CAP and enter cartridges.

Variable: *cap_id* s the CAP used to enter cartridges.

CAP *cap_id*: Unknown media type label.

Explanation: One or more cartridges cannot be entered because the media type is unknown.

Variable: *cap_id* is the CAP used to enter cartridges. Open the CAP and remove the cartridges.

Display Command References

In this chapter you will learn how to use the display command.

If you think in terms of SQL and look at the options, you will realize the power and flexibility in the display command. The display command is a powerful tool for reporting information from the ACSLS database. Like a SQL SELECT statement, display allows you to specify:

- the database table (ACSLS object) you want to display

- wildcards

- conditions that objects must meet to be selected

- fields to report and the order in which they will be reported

- the sort order in which output will be displayed

- that only the count of the number of objects meeting the selection criteria be displayed

The result is a query facility that lets you customize what you want displayed and how it should be presented. Much more information is available through the display command than is possible through ACSLS query commands. For example, `display volume` gives access to volume information that was only available through a custom volrpt. `display lsm` and `display drive` provide LSM and tape drive serial numbers.

The display commands are:

Displays specific CAP information based on the options selected.

Displays specific cell information based on the options selected.

Displays specific drive information based on the options selected.

Displays specific lock information based on user ID.

Displays specific LSM information based on the options selected.

Displays specific library panel information based on the options selected.

Displays specific scratch pool information based on the options selected.

Displays specific port information based on the options selected.

Displays specific volume information based on the options selected.

■ display Commands

The `display` command displays various fields of information for different library components. The display command allows you to select the order and format in which the output is displayed.

The information displayed can include multiple columns and long lines. To prevent lines from wrapping, you can re-size the terminal window used for display. A `cmd_proc -lq` window will probably be your best choice.

Wildcards, numeric or alpha ranges, and selection lists can be used with the display command.

Using Wildcard Characters

- The wildcard character, *, can be used to substitute: for certain display operands; for one or more subfields in compound display operands; or at the beginning or end of a volume ID. It must match one or more characters.

- Many of the ACSLS library components have a primary key composed of multiple fields.

  For example, the primary key of:

  - an LSM is *acs,lsm*

  - a drive is *acs,lsm,panel,drive*

- When you wild card a primary key that contains multiple fields, you can either specify an asterisk (*) for all fields in the key or you can wild card individual key fields. If you specify (*) some of the key fields and wild card others, you must specify all of the key fields.

- Use '_' to match a single character in a *vol_id*. The '_' is positional, and represents a single character. It may be used more than once in a `vol_id` to represent more than one character.

- Examples of valid wild cards using the * character:

  - `display drive *` (display all drives)

  - `display drive 0,1,*,*` (displays all drives in LSM 0,1)

  - `display volume *100` (displays all volumes ending in 100)

- `display drive * -volume RB0001` (displays the drive(s) that contain volume RB0001)

- Examples of invalid wildcard using the * character:

  - `display drive 0,1,*` (Asterisks for both the panel and drive positions must be specified when selecting all drives in LSM 0,1)

  - `display cap * *` (Only one asterisk is allowed when selecting all CAPs)

  - `display volume 1*111` (The embedded asterisk is not allowed. You must use display volume 1_111).

Using Ranges

- A range is denoted as "-".

- Rules for using ranges are:

  - Negative numbers are not allowed, except for volume home location.

  - <n and >n are allowed. <n is all numerics less than n. >n is all numerics greater than n.

  - Multiple numbers or ranges can be used, separated by a white space.

  - Volume ranges can be alphanumeric.

- Examples of valid ranges:

  - `display volume * -drive 0,1,2,2-5` (displays all volumes in acs 0, lsm 1, panel 2, drives 2 through 5)

  - `display lsm * -state online -free_cells >50 -type 9730` (displays all 9730 lsms in an online state with a freecell count greater than 50)

- Examples of invalid ranges:

  - `display drive 0,1,1,1-*` (* is not valid in a range)

  - `display volume * -drive 0,1,?,1-5` (? is not valid in a range)

Format

```
display type arg ... [ -selection ... ] [ [ -c ] | [ -f
field ... ] [ -s sort_field ... ] [ -n n ] ]
```

Options

type
> specifies the object to be displayed. Valid types are: cap, cell, drive, lock, LSMs, panel, pool, port, and volume.

arg
> (argument) specifies the identifiers for the object type. Only objects that match one or more of the identifiers are returned.

-selection
> specifies additional selection criteria. To be selected, a component must match at least one of the selection identifiers for each selection criteria.

-c
> (count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f field

> specifies the fields of information to be returned and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If -f is specified, the default fields are not displayed unless they are key fields or have been specified. If -f is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the -f option.

> The list of fields is separated by spaces.

-s sort_field
> sorts the objects returned by the field(s) selected in the order specified. The default sort order for each display command is discussed under each command. Sorting is based on internal database values and may not always display in alphanumeric order.

> All returned fields for a type are valid sort_fields.

-n n
> specifies the maximum number of objects to be displayed.

■ display cap

The display cap command displays specific CAP information based on the options selected.

Note: SL3000. To prevent CAP IDs from changing when additional modules are added to the SL3000 or additional CAPs are installed, the library returns all possible CAP locations in its configuration information. CAPs that are not installed will be reported as "not installed" in the CAP Status.

ACSLS only adds database records for CAP locations that actually have CAPs installed.

Format

```
display cap cap_id ... ]
[ -availability cap_availability ...]
[ -status cap_status ... ][-priority cap_priority ... ]
[ -state cap_state ... ] [ -manual | -automatic ]
[ -condition cap_condition ... ] [ [ -c ] | [ -f field ... ]
[ -s sort_field ... ] [ -n n ] ]
```

Fields

The fields for display cap are:

- key fields: acs, lsm, cap

- default fields: acs, lsm, cap, status, priority, state, mode, size, availability

- returned fields: acs, lsm, cap, status, priority, state, mode, size, condition, availability

Options

cap_id
> specifies the CAP(s) to display. A cap_id is of the format: *acs*, *lsm*, *cap*.
>
> Wildcard character,*, can be used in any cap_id subfield or used to represent all subfields. However, if it is used to represent all cap_ids (example: *display cap* * or *display cap* *,*,*), additional cap_ids (either numeric or *) are not allowed. Numeric ranges apply to any subfield of cap_id.

availability *cap_availability*
> specifies one or more cap availabilities as additional selection criteria. CAP availabilities are: lib_not_partn, cap_shared, cap_dedicated, dedicated_other, not_installed.
>
> CAPs that are dedicated_other or not_installed are not defined in the ACSLS database. These values are only seen when a CAP's availability has changed (since it was configured). Reconfigure to remove these CAPs.

-status *cap_status*
> specifies one or more CAP statuses as additional selection criteria. CAP statuses are: audit, available, eject, enter, insert_magazines.

-priority *cap_priority*
> specifies one or more CAP priorities. Numeric range rules apply. CAP numeric ranges: 0-16.

-state *cap_state*
>   specifies one or more CAP states. CAP states are: diagnostic, offline, offline_pending, online, recovery.

-manual
>   selects CAPs in manual mode.

-automatic
>   selects CAPs in automatic mode.

-condition *cap_condition*
>   specifies one or more CAP conditions from which to select. Valid conditions are: inoperative, maint_required, or operative.

-c
>   (count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f *field*
>   specifies the fields of information to be returned and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If -f is specified, the default fields are not displayed unless they are key fields or have been specified. If -f is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the -f option.
>
>   The list of fields is separated by spaces.

-s *sort_field*
>   sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.
>
>   All returned fields for a type are valid sort_fields.

-n *n*
>   specifies the maximum number of objects to be displayed.

Examples

- To display all CAP data:

  ```
  display cap *
  ```

- To display CAP data for ACS 1, LSM 1, CAP 1:

  ```
  display cap 1,1,1
  ```

- To display all manual mode CAPs in LSM 1, 0:

  ```
  display cap 1,0,* -manual
  ```

• To display all CAPs in an offline pending state:

```
display cap * -state offline_pending
```

■ display cell

The `display cell` command displays specific information about a cell based on the options selected.

Format

```
display cell-id cell_loc ... [ -status cell_status ... ]
[ [ -c ] | [ -f field ... ] [ -s sort_field ... ]
[ -n n ] ]
```

Fields

The fields for display cell are:

• key fields: acs, lsm, panel, row, column,

• default fields: acs, lsm, panel, row, column, status

• available fields: acs, lsm, panel, row, column, status

Options

-status *cell_status*

specifies one or more cell statuses as additional selection criteria. Valid cell statuses: empty, full, inaccessible, reserved.

The wildcard character,*, can be used in any cell_id subfield or used to represent all subfields. However, if it is used to represent all cell_ids (example: *display cell ** or *display cell_id *,*,*,**), additional cell_ids (either numeric or *) are not allowed. Numeric ranges apply to any subfield of cell_id.

-c

(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f *field*

specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

The list of fields is separated by spaces.

-s *sort_field*
    sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.

    All returned fields for a type are valid sort_fields.

-n *n*
    specifies the maximum number of objects to be displayed.

Example

To display information for cells with a reserved status:

```
display cell * -status reserved
```

■ display drive

The `display drive` command displays specific drive information based on the options selected.

Format

```
display drive drive_id ... [ -status drive_status ... ]
[-state drive_state ... ] [ -type drive_type ... ]
[ -volume vol_id ... ] [ -lock lock_id... ] [ -serial
drive_serial_num ... ] [ -condition drive_condition ...]
[ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

Fields

The fields for display drive are:

- key fields: acs, lsm, panel, drive

- default fields: acs, lsm, panel, drive, status, state, volume, type

- available fields: acs, lsm, panel, drive, status, state, volume, type, lock, serial_num, condition, wwn, last_dismount_time, error, end_of_life, load_limit_alert

ACSLS can only report drive serial numbers when the drive reports its serial number to the library and the library reports its serial number to ACSLS.

Only modern libraries and drives report the drive serial numbers to ACSLS. This includes T9840, T9940, LTO, DLT 7000, SDLT, and later drives.

SCSI-attached libraries, L5500, and later libraries report serial numbers to ACSLS.

Options

drive_id
　　displays the drive_id in the following format: *acs*, *lsm*, *panel*, *drive*.

　　Wildcard character,*, can be used in any drive_id subfield or used to represent all subfields. However, if it is used to represent all drive_ids (example: *display drive* * or *display drive_id* *,*,*,*), additional drive_ids (either numeric or *) are not allowed. Numeric ranges apply to any subfield of drive_id.

-status *drive_status*
　　displays one of the following statuses as additional selection criteria: available, in_use, reserved

　　Possible scenario: You perform a *query drive all* and get a message saying the drive is in use. You then perform a *display drive* * and get a message saying the drive is reserved. This means: the reserved status for the drive implies that a mount request is in progress and that a volume is on its way to the drive. At the same time the drive is considered to be in use.

-state *drive_state*
　　displays the following states for one or more drives: diagnostic, online, offline, recovery.

-type *drive_type*
　　displays drives by transport type.

-volume *vol_id*
　　displays volume(s) mounted on the drive(s) selected. A vol_id can be a 6-alphanumeric string or wildcard string that specifies a tape volume or range of volume IDs.

-lock *lock_id*
　　displays drive information on locked drives by lock ID.

-serial *drive_serial_num*
　　specifies the drive serial number.

-condition *drive_condition*
　　displays condition of the drive(s) specified. Valid conditions are: operative, inoperative, or maint_required.

-c
　　(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f *field*
　　specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If -f is specified, the default fields are not

displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

The list of fields is separated by spaces.

The drive's end_of_life value is four digits that represent the drive's use as a percentage of its end_of_life setting. The four digits express a value from 000.0% to over 100.0%.

The drive's load_limit_alert is a boolean value, where: True=1 and False=0. When a drive that reports an end_of_life percentage reaches 100%, its load_limit_alert is set to 1. Other drives only report a load_limit_alert.

`-s` *sort_field*
    sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.

    All returned fields for a type are valid sort_fields.

`-n` *n*
    specifies the maximum number of objects to be displayed.

Examples

- To display all drives in panel 0,1,1.

  ```
  display drive 0,1,1,*
  ```

- To display all drives in LSM 0,1.

  ```
  display drive 0,1,*,*
  ```

- To display a maximum of 56 rows for all drives in ACS 1, sorted by panel and drive:

  ```
  display drive 1,*,*,* -s panel drive -n 56
  ```

- To display drive data for ACS 1, LSM 1, with status available:

  ```
  display drive 1,1,*,* -status available
  ```

- To display drive types, serial numbers, and world wide names (wwn) for all drives:

  ```
  display drive * -f type serial_num wwn
  ```

- To display drives sorted by end_of_life percentage:

  ```
  display drive * -f type end_of_life -s end_of_life
  ```

■ display lock

The `display lock` command displays specific lock information by *lock_id* based on the options selected.

Format

```
display lock lock_id ... [ -user user_id ... ] [ [ -c ] |
[ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

Fields

The fields for display lock are:

- key fields: lock_id

- default fields: lock_id, user_id

- available fields: lock_id, user_id

Options

lock_id

displays the numeric lock_id.

Wildcard character,*, can be used in any lock_id subfield or used to represent all subfields. However, if it is used to represent all lock_ids (example: *display lock ** or *display lock_id *,*,**), additional lock_ids (either numeric or *) are not allowed. Numeric ranges apply to any subfield of lock_id.

user_id

specifies the user name (example: acsss) that has locked the volume or transport. The *user_id* wraps after 80 characters.

-c

(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f *field*

specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If -f is specified, the default fields are not displayed unless they are key fields or have been specified. If -f is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the -f option.

The list of fields is separated by spaces.

-s *sort_field*
>   sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.
>
>   All returned fields for a type are valid sort_fields.

-n *n*
>   specifies the maximum number of objects to be displayed.

Example

To display lock information for lock_id 2:

```
display lock 2
```

■ display lsm

The `display lsm command` displays specific LSM information based on the options selected.

Format

```
display lsm lsm_id ... [ -status lsm_status ...  ]
[-state lsm_state ... ] [ -free_cells cell_count ... ]
[ -type lsm_type ... ] [ -serial lsm_serial_num ... ]
[ -condition lsm_condition ] [ -door_open | -door_closed ]
[ [ -c ] | [ -f field ... ] [ -s sort_field ... ]
[ -n n ] ]
```

Fields

The fields for display lsm are:

- key fields: acs, lsm

- default fields: acs, lsm, status, state, free_cells

- available fields: acs, lsm, status, state, free_cells, type, serial_num, condition, door_status

lsm_id
>   displays the lsm id in the following format: *acs*, *lsm*.
>
>   Wildcard character, *, can be used in any lsm_id subfield or used to represent all subfields. However, if it is used to represent all lsm_ids (example: *display lsm ** or *display lsm_id *,**), additional lsm_ids (either numeric or *) are not allowed. Numeric ranges apply to any subfield of lsm_id.

-status *lsm_status*
    specifies one or more LSM statuses from which to select. Valid statuses
    are: audit, normal

-state *lsm_state*
    specifies one or more LSM states from which to select. Valid states are:
    diagnostic, online, offline, offline_pending, recovery.

-free_cells *cell_count*
    selects LSMs by the number of free cells.

-type *lsm_type*
    displays LSMs by type.

-serial *lsm_serial_num*
    displays serial numbers for selected LSMs.

-condition *lsm_condition*
    displays condition of the selected LSMs. Valid conditions are: operative,
    inoperative, or maint_required.

-door_open
    displays LSMs with open doors.

-door_closed
    displays LSMs with closed doors.

-c
    (count) causes the request to display only the number of objects that
    satisfy the *arg* and *selection* criteria. This option is mutually exclusive with
    the -f field, -s sort_field, and -n *n* options.

-f *field*
    specifies the fields of information and the order in which they are to be
    returned for each object selected. Each display type has a list of available,
    key, and default fields. If -f is specified, the default fields are not
    displayed unless they are key fields or have been specified. If -f is not
    specified, then the default fields are returned. Key fields for each type are
    normally displayed first. However, you can change the order in which the
    key fields are displayed by specifying them after the -f option.

    The list of fields is separated by spaces.

-s *sort_field*
    sorts the objects returned by the field(s) selected in the order specified.
    Sorting is based on internal database values and may not always display
    in alphanumeric order.

    All returned fields for a type are valid sort_fields.

-n *n*
    specifies the maximum number of objects to be displayed.

Example

To display information for all LSMs in the library for type 9714:

```
display lsm * -type 9714
```

To display types all LSMs:

```
display lsm * - f type
```

◼ display panel

The `display panel` command displays specific library panel information based on the options selected.

Format

```
display panel panel_id ...[ -type panel_type ... ] [ [ -c ] |
[ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

Fields

The fields for display panel are:

- key fields: acs, lsm, panel

- default fields: acs, lsm, panel, type

- available fields: acs, lsm, panel, type

Options

panel_id
    displays the panel id in the following format: *acs*, *lsm*, *panel*.

    Wildcard character, *, can be used in any panel_id subfield or used to represent all subfields. However, if it is used to represent all panel_ids (example: *display panel* * or *display panel_id* *,*,*), additional panel_ids (either numeric or *) are not allowed. Numeric ranges apply to any subfield of panel_id.

`-type` *panel_type*
    specifies one or more valid panel types.

`-c`
    (count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f *field*

    specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If -f is specified, the default fields are not displayed unless they are key fields or have been specified. If -f is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the -f option.

    The list of fields is separated by spaces.

-s *sort_field*

    sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in numeric order.

    All returned fields for a type are valid sort_fields.

-n *n*

    specifies the maximum number of objects to be displayed.

Example

Display panel data for all panels:

```
display panel *
```

■ display pool

The display pool command displays specific scratch pool information based on the options selected.

Format

```
display pool pool_id ... [ -low_water low_water_mark ... |
-high_water high_water_mark... ]
[-overflow | -no_overflow ] [ [ -c ] | [ -f field ... ]
[ -s sort_field ... ] [ -n n ] ]
```

Fields

The fields for display pool are:

- key fields: pool_id

- default fields: pool_id, low_water, high_water, overflow

- available fields: pool_id, low_water, high_water, overflow

Options

pool_id
> displays the pool id of a scratch pool.
>
> Wildcard character, *, can be used in any pool_id subfield or used to represent all subfields. However, if it is used to represent all pool_ids (example: *display pool* * or *display pool_id* *,*,*), additional pool_ids (either numeric or *) are not allowed. Numeric ranges apply to any subfield of pool_id.

-low_water *low_water_mark*
> specifies the minimum number of scratch cartridges in the pool. If a low_water mark is displayed with a hyphen (-) appended to it, the number of cartridges in the library for the selected scratch pool is within the designated low_water mark.

-high_water *high_water_mark*
> specifies the maximum number of scratch cartridges in the pool. If a high_water mark is displayed with a hyphen (-) appended to it, the number of cartridges in the library for the selected scratch pool is within the designated high_water mark.

-overflow
> selects pools wherein if a scratch mount is requested and if the pool is empty, the common pool is used to satisfy the request.

-no_overflow
> selects pools wherein if a scratch mount is requested and if the pool is empty, the mount fails without checking the common pool for available tapes.

-c
> (count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f *field*
> specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If -f is specified, the default fields are not displayed unless they are key fields or have been specified. If -f is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the -f option.
>
> The list of fields is separated by spaces.

-s *sort_field*
>   sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in numeric order.
>
>   All returned fields for a type are valid sort_fields.

-n *n*
>   specifies the maximum number of objects to be displayed.

Example

- To display all information for pool 5:

  `display pool 5`

- To display all pools with overflow:

  `display pool * -overflow`

■ display port

The `display port` command displays specific port information based on the options selected.

Format

```
display port port_id ... [ -online | -offline ]
[ -name port_name ... ] [ [ -c ] | [ -f field ... ]
[-s sort_field ... ] [ -n n ] ]
```

Fields

The fields for display port are:

- key fields: acs, port

- default fields: acs, port, name, state

- available fields: acs, port, name, state, desired_state

Options

port_id
>   displays the port id in the following format: *acs*, *port*.
>
>   Wildcard character, *, can be used in any port_id subfield or used to represent all subfields. However, if it is used to represent all port_ids (example: *display port* * or *display port_id* *,*,*), additional port_ids (either

numeric or *) are not allowed. Numeric ranges apply to any subfield of port_id.

- `online`

  selects and displays ports in an online state.

- `offline`

  selects and displays ports in an offline state.

- `name` *port_name*

  specifies the port name.

`-c`

  (count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

`-f` *field*

  specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

  The list of fields is separated by spaces.

  The port's desired_state is the state you want the port to be in (as a result of vary commands).

`-s` *sort_field*

  sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order

  All returned fields for a type are valid sort_fields.

`-n` *n*

  specifies the maximum number of objects to be displayed.

Example

To display all ports in an online state:

```
display port * -online
```

■ display volume

The `display volume` command displays specific volume information based on the options selected.

Format

```
display volume vol_id ... [ -home acs,lsm,panel,row,column...]
[ -drive drive_loc ... ] [-data | -scratch | -clean ]
[ -media media_type ... ] [ -pool pool_id... ] [ -standard |
-virtual ] [ -status vol_status ... ] [ -entry entry_date ... ]
[ -access  access_date ... ] [ -lock lock_id ... ] [ [ -c ] |
[ -f field ... ] [ -s sort_field ... ]  [ -n n ] ]
[ -max_use max_use] [ -lock_time lock_time]
```

Fields

The fields for `display volume` are:

- key field: vol_id

- default fields: vol_id, acs, lsm, panel, row, column, pool, status, media, type

- available fields: vol_id, acs, lsm, panel, row, column, drive, type (displayed as lsm,panel,drive), media, pool, label_type, status, entry_date, access_date, access_count, max_use, lock, lock_time, recording_format_family, recording_format_model, encrypt_status, volsafe_status, media_status, warranty_life, end_of_life, load_limit_alert

Options

`vol_id`
  displays the volume(s) selected.

  A vol_id can be one or more 1-6 alphanumeric strings that specifies a tape cartridge.

  The wildcard character, *, can be used to represent all vol_ids (example: *display vol **) or to represent a group of cartridges (example:.*display vol_id VAP** displays all cartridges starting with VAP).

  The *vol_id* may contain any valid volume ID characters and *.

`-home` *home_loc*
  specifies volume home location(s) from which to retrieve volume information in the following format: *acs,lsm,panel,row,column*. It cannot be used with *-drive*.

  home_loc subfields can use numeric ranges. All *home_loc* subfields must be specified. A wildcard character, *, can be used in place of one or more *drive_loc* subfields.   Numeric range rules apply to any subfield or drive_loc.

`-drive` *drive_loc*

specifies drive location(s) from which to retrieve volume information in the following format: *acs*,*lsm*,*panel*,*drive*. It cannot be used with *-home*

drive_loc subfields can use numeric ranges. All *home_loc* subfields must be specified. A wildcard character, *, can be used in place of one or more *drive_loc* subfields. Numeric range rules apply to any subfield or drive_loc.

`-data`

selects data cartridges.

`-scratch`

selects cartridges in scratch mode.

`-clean`

selects cleaning cartridges.

`-media` *media_type*

selects only cartridges with the specified media type.

`-pool` *pool_id*

selects and displays cartridges from scratch pools. Cartridges retain the ID of the scratch pool even after they have changed to data cartridges.

`-standard`

selects a tape(s) with a scannable volume ID label_type. It cannot be used with *-virtual*.

`-virtual`

selects a tape(s) without an external volume ID label_type that was entered through the venter command. It cannot be used with *-standard*.

`-status` *vol_status*

valid entries are: *dismount*, *eject*, *enter*, *mount*, *home*, *in_drive*, *move*, *missing*, *absent*, *ejected*.

`-entry` *entry_date*

specifies the date and time the cartridge was entered into the library. It is displayed in ISO standard format.

`-access` *access_date*

specifies the date and time the cartridge was last used. It is displayed in ISO standard format.

`-lock` *lock_id*

specifies the lock ID for the volume.

`-max_use` *max_use*

specifies the maximum number of uses for a cleaning cartridge. Non-cleaning cartridges have a max_use value of zero.

`-lock_time` *lock_time*

specifies the date and time the lock was set. It is displayed according to the system default format.

-c

(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the -f field, -s sort_field, and -n *n* options.

-f *field*

specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If -f is specified, the default fields are not displayed unless they are key fields or have been specified. If -f is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the -f option.

The list of fields is separated by spaces.

The volume's end_of_life percentage is four digits that represent the volume's use as a percentage of its end_of_life setting. The four digits express a value from 000.0% to over 100.0%.

The volume's load_limit_alert is a boolean value, where: True=1 and False=0. When a volume that reports an end_of_life percentage reaches 100%, its load_limit_alert is set to 1. Other volumes only report a load_limit_alert.

A volume's warranty_life is also a percentage (000.0% to over 100.0%) of its warranty usage life.

-s *sort_field*

sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order

All returned fields for a type are valid sort_fields.

-n *n*

specifies the maximum number of objects to be displayed.

Examples

- To display volumes based on location:

  This example displays all volumes with home locations in panel 0,1,9.

  `display volume * -home 0,1,9,*,*`

- To display all 9840 data volumes.

  `display volume * -media STK1R`

- To display 9840 data volumes with their recording formats (if known):

```
display volume * -media STK1R -f media
recording_format_family recording_format_model
```

- To display a cleaning volumes by ACS and media type:

```
display volume * -home acs,*,*,*,* -media media_type
-f vol_id acs lsm media max_use access_count
```

- To display volumes by date:

  - To display all volumes entered this year:

    ```
    display volume * -entry 2002
    ```

  - To display all volumes entered in January, 2002

    ```
    display volume * -entry 2002-01
    ```

  - To display all volumes entered on January 27, 2002

    ```
    display volume * -entry 2002-01-27
    ```

  - To display all volumes entered on January 27, 2002 between 10:00 a.m. and 11:00 a.m.

    ```
    display volume * -entry 2002-01-27:10
    ```

  - To display all volumes entered between January 2002 and April 2002.

    ```
    display volume * -entry 2002-01-2002-04
    ```

  - To display all volumes entered between 6:33 a.m. and 6:57:45 April 10 (for example, you may want to do this after an outage).

    ```
    display volume * -entry 2002-04-10:06:33:00- 2002-
    04-10:6:57:45
    ```

  - To display all volumes entered since February 25, 2002.

    ```
    display volume * -entry >2002-02-25
    ```

  - To display T10000 data volumes sorted by end_of_life usage:

    ```
    display volume * -media T10000T1 T10000TS -f media
    end_of_life warranty_life -s end_of_life
    ```

ACSLS Backup and Recovery Tools

<div style="text-align:right">**A**</div>

■ ACSLS Backup Tools

ACSLS offers three robust and distinctly different methods for backing up both its database and ACSLS control files. Each utility performs different functions and all methods play an important role in a complete disaster recovery plan.

This appendix:

- Outlines and explains each utility, what they are used for, and why they are important.

- Provides a high level view of disaster recovery scenarios

Automatic Backups

ACSLS provides automated database protection services. These automated protection services safeguard the daily operation of the ACSLS database against changes that can produce either unintended consequences or from database corruption

As a result of these automated backup protection services, you have the ability to restore your database back to any backup time from the present to the end of your retention period. The restore tools are discussed later in this appendix.

This section discusses the automated backup methods and why they are used.

- ACSLS default backup directory

  During the initial installation of ACSLS, you were asked to supply the name of the directory to use for backups (`/export/backup` by default). It is in this directory that backup activity occurs.

  - A complete database backup is performed and placed in the directory, using a date naming convention:

    `/export/backup/y`*`yyy-mm-dd-hh:mm:ss.tar`*.

    The time at which the daily backup is performed can be modified by changing the "Automatic Backup Variables" within `acsss_config`.

Refer to "Setting Variables that Control ACSLS Behavior" on page 31 for information on changing default backup behavior.

- Database retention period

Another configurable parameter within ACSLS that affects automatic backups is the database retention period. This is defined as the amount of time ACSLS retains the backups.

The default for the retention period is 8 days.

Refer to "Setting Variables that Control ACSLS Behavior" on page 31 for information on changing default backup behavior.

The retention period can also be modified by using `acsss_config.`

Manual Backups

ACSLS provides a utility called `bdb.acsss` that backs up the critical ACSLS data using the command line. This is also the method used to restore the ACSLS database where the environment consists of the same or identical hardware, OS level, and ACSLS version. Refer to "bdb.acsss" on page 131.

Used without any command line options, *bdb.acsss* provides the ability to create a database backup and store it in the default backup directory. All critical ACSLS database and ACSLS control files are backed up to a single file. This file can then be used to restore ACSLS to its previous state on the same or identical hardware for scenarios such as an internal disk or motherboard failure.

The `rdb_acsss` utility allows a "-f" option that can be used to either specify a file and location (`rdb.acsss -f /path/my_file`) or a tape device (`-f /dev/rmt/0mn`). When using a tape device, you do not provide a file name, on the tape device.

Manual Database Exports

ACSLS provides a utility called `db_export.sh` to export the ACSLS database, ACSLS control files, and any customized dynamic variables. The `db_export.sh` utility is responsible for dumping the ACSLS database to comma separated ACSII files, making a copy of the ACSLS control files, and making a copy of the dynamic variables. This is the method used to migrate to newer versions of ACSLS and is not recommended for daily backup operations because both ACSLS and the database must be down before performing the export.

The `db_export.sh` command line utility is the preferred method for migrating the database between different levels of server hardware, OS versions, and different releases of ACSLS. Without options, it can be used with the local default tape device such as `/dev/0mn`. This tape can then be

moved to any location, and ACSLS and its associated ACSLS control files can be restored into any OS version or level of ACSLS.

Note: Although any tape device can be selected, a **no-rewind** device should be used. The `db_export` utility creates two files. If a rewind device is selected, the first file (datafiles) would be overwritten when the second file is created.

As in the `bdb.acsss` utility, the "`-f`" option can be used to specify a tape device other than the system default. Simply execute `db_export.sh /dev/0mn` or any attached tape device to use this option.

The `-f` option also allows the database to be exported to the named file. When using this method, you will notice that two files are created, the one that you named and also another file with a `.misc` extension. Both files must be transferred to the server where the import will take place to ensure a successful import.

When executing the `db_export.sh` utility either with the `-f` option or without, you will be prompted to choose the version of ACSLS to which you are exporting. Selections are:

```
1: ACSLS 5.3.2 or 5.4

2: ACSLS 6.0 or 6.0.1

3: ACSLS 6.0.1 with L700e

4: ACSLS 6.0.1 with PUT0201

5: ACSLS 6.1, 7.0, 7.1/7.1.1 before PUT0701

6: ACSLS 7.1/7.1.1 with PUT0701 or ACSLS 7.2 (any)

7: ACSLS 7.3 or later

E: Exit
```

Choose the appropriate version.

■ ACSLS Recovery Tools

ACSLS uses two different recovery tools to restore all backups and exports. Both offer a menu driven user interface and easily selectable options. These two utilities are:

- rdb.acsss - the recovery tool for both automated and manual backups

- db_import.sh - restores an exported database and/or ACSLS control files from the same version of ACSLS, a different version of ACSLS, or from a different hardware platform. This option also allows the recovery of customized dynamic variables.

Using rdb.acsss

The `rdb.acsss` utility restores the ACSLS database and the ACSLS Control files using a backup created by either the automatic backup function or the `bdb.acsss` utility. The ACSLS Control files are located in $ACS_HOME/data, and define several different environmental variables for ACSLS. They specify Access Control settings, scratch preferences, Extended Store LSMs, custom `volrpt` settings, and volume attributes (for `watch_vols` utility), etc

Refer to "rdb.acsss" on page 166 for options and procedures.

Using db_import.sh

ACSLS provides a `db_import.sh` utility to restore an exported database from the same version of ACSLS, a different version of ACSLS, or it could even be from a different hardware platform. Like `rdb.acsss`, it offers an easy to read menu driven user interface allowing you to select the task that you wish to perform.

The `db_import.sh` utility can work without options or you can supply the "`-f`" option with a path and file name as an argument. Executing `db_import.sh` from the command line without any options causes the utility to look for the exported database on the local tape device. It first checks for the existence of the exported database, verifies that it is a valid database export file, and then displays a menu with four options.

Note: You can also supply a -f option with a tape device (`-f /dev/rmt/0mn`) for a non-default device. Although you can supply any valid tape device, it is a requirment that you supply a **no-rewind** device. The db_import.sh utililty uses two files, one for data and one for control files. If you use a rewind device, after data files are recovered, the tape would be rewound and the control files would fail.

If you provide the `-f` option with a path and file name, db_import.sh uses the supplied file name as the exported database file. As with the local tape device, it first checks to see if the file exists and then validates that the supplied file name is an exported database file. If the supplied file is a valid export, it displays a menu. The menu options are:

• Option 1 - Import database only from either a different platform or from an earlier release

Whether you are recovering the database from a local tape device or from a file, this option initiates an import of the data only and does not recover any ACSLS control files. This is the desirable method for migrating from one platform to another or if you don't want customizes files from a previous installation.

- Option 2 - Import customizable configuration files (`$ACS_HOME/data/external` only) exported from a different platform or from an earlier release.

  These files would include any access control information or customized volume report templates.

- Option 3 - Recover a previous environment on the same platform and version (Disaster Recovery) - database, `$ACS_HOME/data/external` and internal

  This option is used to recover or rebuild an identical server from the one from which the data was exported. This can be used in situations where you are retiring a server and building a new one. You are able to restore the ACSLS database and ACSLS control files to exactly the same state that was exported.

- Option 4 - Merge customized dynamic variables from a previous installation

  Even if you are unsure if you have any customized variables, it is a good idea to run this option during migration because it will not change any unchanged variables.

◼ Disaster Scenarios

Database becomes corrupted – need to restore the database to a working state

1. As user "acsss", idle and stop ACSLS and the database.

   >idle   (from cmd_proc)
   $ kill.acsss   (from UNIX command line)
   $ db_command stop (from UNIX command line)

2. Start `rdb.acsss` utility

3. Select option 1. Refer to .

4. When the recovery is complete, start ACSLS and test

Ran acsss_config against the wrong library

1. Select option 2. Refer to .

2. Start ACSLS and test according to procedures in *"Database Backup and Restore"* chapter.

Server failure – rebuilding the same server with new hardware

1. Install the operating system.

2. Configure the new server and OS with the settings from the previous server.

3. Install ACSLS.

4. Insert the backup tape or FTP backup file onto the server.

5. Start the `rdb.acsss` utility.

6. Select option 2. Refer to .

7. Exit `rdb.acsss`.

8. Start ACSLS and test according to procedures in *Database Backup and Restore*.

Server failure – rebuilding another ACSLS server with new hardware

1. Install the operating system.

2. Install ACSLS.

3. Place the server-to-server backup files in the proper location.

4. Enter `rdb.acsss`. Refer to

5. Select option 3.

6. When the recovery utility completes, start ACSLS and test according to procedures in *Database Backup and Restore*.

   Note:

ACSLS Support of the SL8500

<div style="text-align: right;">**B**</div>

This appendix discusses the following topics:

Note: To support library partitioning with ACSLS 7.1, you must install PUT0701. Refer to "Library Partitioning" on page 367 for more information.

This appendix assumes a minimum of SL8500 firmware level 2.5.2. Warnings and procedures for earlier levels of SL8500 firmware have been removed.

The following figure shows the SL8500 library with an ACSLS server.

Figure 17. SL8500 with an ACSLS Server
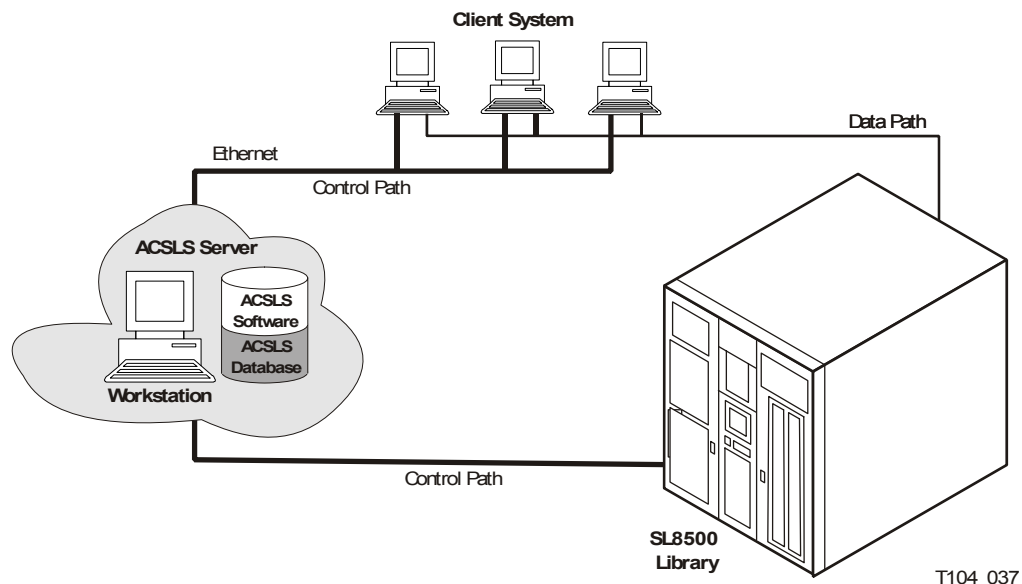
Before Configuring ACSLS for the SL8500

Before you configure ACSLS for the SL8500:

- Connect one or more SL8500s to ACSLS

- Verify that all the components of the SL8500 are operational.

    ACSLS builds its library configuration from the information reported by the library. If SL8500 components are not operational, the library information may not be reported to ACSLS, and the ACSLS configuration of the SL8500 will be incomplete.

    Note: If the components like drives or CAPs are not operational, it is easy to use Dynamic Configuration (`config acs`, `config lsm`, or `config drives`) to add or update them while ACSLS is running and the library is online.

Connecting to Multiple SL8500s in an ACS

The SL8500 library uses TCP/IP protocol over an Ethernet physical interface to manage and communicate with the host and ACSLS. This interface enables ACSLS to connect to and communicate with the SL8500.

When SL8500 3.97 or higher firmware is installed, ACSLS can connect to more than one SL8500 in an ACS (library complex).

ACSLS supports up to four connections to an ACS. For example, this can be: four connections to four SL8500s; two connections to each of two SL8500s;

two connections to one SL8500 and two connections to two other SL8500s; three connections to two or three libraries, etc.

When ACSLS is connected to more than one library, the connections should be through different subnets for redundancy. If one subnet fails, communication between ACSLS and the library still continues over the other subnet(s).

When ACSLS has two connections to one SL8500, you must configure the SL8500 and ACSLS server routing tables as described in "ACSLS Dual TCP/IP Support " on page 344. If you have only a single connection between the ACSLS server and each SL8500 library, configuring the ACSLS and SL8500 routing tables is **not** necessary.

To optimize library performance and minimize inter-library communication among SL8500s, connect to the libraries with the most activity. Make the first connection that you specify in `acsss_config` or `config acs new` to the SL8500 with the most activity.

For more information, refer to the *SL8500 Modular Library System Technical Brief - Host to Library Communications*.

Verifying that all SL8500 Components are Operational

To verify that all the components of the SL8500 are operational:

1. Logon to the StorageTek Library Console™ (SLConsole).

   You can use either the console on the SL8500 or a remote Library Console.

2. Select `Tools -> System Detail`.

   • All SL8500 components should be green.

   Exception: Drives that are yellow can be configured now, or later, using dynamic configuration ("config drives" on page 137).

   • Missing components can be added using the Dynamic Configuration (`config acs` or `config lsm`) utility.

   • **IMPORTANT**: Before configuring the SL8500, the elevators (Elevator Folder) **must** be green. If the elevators are not green, **do not** configure the SL8500 to ACSLS. The elevators are the logical pass-thru-ports (PTPs). Without PTPs, ACSLS will not know that the SL8500 rails are connected.

3. Once the SL8500 components are operational, configure SL8500 to ACSLS, as discussed in "Configuring or Updating your Library Hardware" on page 46 or under the "Dynamic Configuration (config) utilities" on page 133.

■ Understanding SL8500 Internal Addresses and ACSLS Addresses

There are differences in the internal addresses of the SL8500 and other libraries supported by ACSLS and HSC.

- The SL8500 is *one's*-based (1) and uses negative numbers.
- Other libraries use a *zero*-based (0) with *no* negative numbers.
- The SL8500 uses *five* parameters: library, rail, column, side, and row.
- Legacy StorageTek libraries (e.g. 9310) use: ACS, LSM, panel, row, and column (HLI–PRC).

Table 26. Addressing Descriptions

HLI–PRC	SL8500	Description	
ACS	**Library**	Number of the specific SL8500 library in a library complex. An ACS is a SL8600 library complex. There can be multiple SL8500s in a library complex.	
LSM LSM 0 ⇨ LSM 1 ⇨ LSM 2 ⇨ LSM 3 ⇨	**Rail** Rail 1 Rail 2 Rail 3 Rail 4	The SL8500 library has four *rails* that the HandBots travel, which are numbered from top to bottom **1**–4 (one's-based). ACSLS considers *each rail* to be a separate *LSM*, numbered from top to bottom **0**–3 (zero-based).	
Panel Panel 0 ⇨ Panel 1 ⇨ Panel 2–*n* ⇨	**Column** CAP Drives Storage slots	Columns indicate the *horizontal* location in the library. As viewed from the front of the library column and panel numbers start at the *center* of the drive panel (1) and sweep forward with increasing numbers. **Note:** The SL8500 does not use panels as an address. An HLI panel spans across the width of the library to include both sides (left and right) and both walls (inner and outer) for *each* LSM.	
	Side	Wall location 1. Outer wall 2. Inner wall	HandBot number 1. Left (–) 2. Right (+)

Table 26. Addressing Descriptions

Row	Row	Rows indicate the *vertical* location of a tape cartridge and are numbered from the top—down.	
Column		Rows for the **HLI** address are: • Storage panels start at 2 with Column 0 = left and Column 1 = right • Rows **0**–12 outer walls • Rows 13–26 inner walls • Each column in a normal storage panel has 27 rows. • For a total capacity of 54 cartridges per panel.	Rows for the **SL8500** address are: • Storage slots start at Column -3 = left Column +3 = right • Rows **1**–13 outer wall • Rows **1**–14 inner wall
• Zero-based numbering (as with HLI) starts numbering at 0. • One's-based numbering (as with the SL8500) starts numbering at 1. • This is an important difference in the numbering sequences between software (ACSLS or HSC) and hardware (physical SL8500 addresses)			

■ Using SLConsole to Translate Addresses

You can use the StorageTek Library Console™ (SLConsole) Search utility to translate between the SL8500 internal address and the ACSLS or HSC panel, row, and column. To locate a cartridge:

1. Log in to the SLConsole.

2. Select Diagnostics -> Search.

3. Select Location.

4. Select one of the following operations in the Location field:

contains	Example: 1,1,-9 lists the contents in Library 1, Rail 1, Column -9 for all rows on both sides
endsWith	Example: 1,5 lists the slot contents for all rails and columns for Side 1, Row 5
equals	Example: 1,1,-9,1,1 lists the contents in that specific location (L,R,C,S,W)
startsWith	Example: 1, 3 lists the slot contents for all columns, sides, and rows in Library 1, Rail 3

5. Select one of the following from the Requestor pull-down menu.

   • default

   The physical location inside the library (cell, drive, CAP).

If you know the physical location (the internal address), and need to find the HLI-PRC address, enter that address in the *location* and pick *default* as the requester.

- hli1

  This is the HLI-PRC address of the cartridge from the library management software.

  If you know the HLI-PRC address and want to find the physical location (internal address) enter that address in the *location* and pick *hli1* as the requester.

  This option displays both the internal address and the hli1 Requester address as shown in the following example.

6. Click the Search button in the top right corner of the SLConsole.

   The search result lists the location by slot-type (cell, drive, or CAP).

7. Click the *Details* button for more information such as: VOLID: media and cartridge type (LTO, SDLT tape, and T-Series; data, cleaning, or diagnostic); and HLI address for cartridges when you specify a *default Requester*.

Tape Drive Locations

The tape drives are associated with and belong to an *LSM*. To mount a cartridge tape in a different LSM, the cartridge must go through an internal pass-thru operation (in this case, *the elevator*) to the drive.

The following tables shows the **internal**—software—mapping (viewed from looking *inside* the library at the tape drives), and

the **external**—physical—numbering of the drives (looking *outside* at the rear of the Drive and Electronics Module).

Table 27. Software Drive Numbering

LSM	ACSLS — Drives Numbers			
Rail 1 LSM 0	**Drive 0**	Drive 4	Drive 8	Drive 12
	Drive 1	Drive 5	Drive 9	Drive 13
	Drive 2	Drive 6	Drive 10	Drive 14
	Drive 3	Drive 7	Drive 11	Drive 15
Rail 2 LSM 1	Drive 0	Drive 4	Drive 8	Drive 12
	Drive 1	Drive 5	Drive 9	Drive 13
	Drive 2	Drive 6	Drive 10	Drive 14
	Drive 3	Drive 7	Drive 11	**Drive 15**
Rail 3 LSM 2	Drive 0	Drive 4	Drive 8	Drive 12
	Drive 1	Drive 5	Drive 9	Drive 13
	Drive 2	Drive 6	Drive 10	Drive 14
	Drive 3	Drive 7	Drive 11	Drive 15
Rail 4 LSM 3	Drive 0	Drive 4	Drive 8	Drive 12
	Drive 1	Drive 5	Drive 9	Drive 13
	Drive 2	Drive 6	Drive 10	Drive 14
	Drive 3	**Drive 7**	**Drive 11**	**Drive 15**

Table 28. Physical Drive Numbering

External—Physical Drive Numbers			
Drive 61	Drive 62	Drive 63	**Drive 64**
Drive 57	Drive 58	Drive5 9	Drive 60
Drive 53	Drive 54	Drive 55	Drive 56
Drive 49	Drive 50	Drive 51	Drive 52
Drive 45	Drive 46	Drive 47	Drive 48
Drive 41	Drive 42	Drive 43	Drive 44
Drive 37	Drive 38	Drive 39	Drive 40
Drive 33	Drive 34	Drive 35	Drive 36
Drive 29	Drive 30	Drive 31	Drive 32
Drive 25	Drive 26	Drive 27	Drive 28
Drive 21	Drive 22	Drive 23	Drive 24
Drive 17	Drive 18	Drive 19	**Drive 20**
Drive 13	Drive 14	Drive 15	Drive 16
Drive 9	Drive 10	Drive 11	Drive 12
Drive 5	Drive 6	Drive 7	Drive 8
Drive 1	**Drive 2**	**Drive 3**	**Drive 4**

These tables show a matching of drives (the highlighted drives). *For example:*

• Internal/software LSM 0 Drive 0 matches with external/physical Drive 64.
• Internal LSM 1 Drive 15 matches with external/physical Drive 33.
• Internal LSM 2 Drive 3 matches with external physical Drive 20.

■ SL8500 CAP Behavior

SL8500 CAPs span three rails (2-4). For ACSLS, this is LSMs 1-3. The SL8500 has one CAP with an optional Dual CAP. Each CAP has three magazines with 13 cells per magazine. Each magazine is adjacent to a separate rail, and can only be accessed by the handbots on that rail.

When the Dual CAP is not present, ACSLS reports that the CAP is offline or non-operational. To prevent any messages regarding the CAP from being displayed, *vary* the non-existent CAP to an offline state using the `vary` command. This also prevents any mistaken attempts to use the Dual CAP where it does not exist.

The SL8500 CAP is treated as an independent component, and does not belong to any one rail (LSM).

Examples:

- The SL8500 CAP can be online and operational even though the LSM identified in the CAP ID is offline or not operational.

- If LSM 1 is varied offline, the SL8500 CAP(s) are not automatically varied offline.

- If LSM 1 is varied online, ACSLS will not automatically vary the SL8500 CAP(s) online.

- The SL8500 CAP(s) can be online and used to enter or eject cartridges to and from SL8500 LSM 0, even though some of the SL8500 LSMs 1-3 are offline and/or not operational.

Note: For CAP behavior in a partitioned library, refer to .

Enter or Eject Operations

During an enter, ACSLS will always try to move the cartridge to an LSM (rail) adjacent to the CAP magazine. For ejects, ACSLS will always try to eject cartridges to a CAP cell adjacent to the LSM containing the cartridge

If these two operations are not possible, the library controller takes care of moving the cartridge through the elevator to another LSM. This requires movements between two handbots and the elevator.

Enter, Eject, and Audit Operations for Some ACSLS Clients

Unlike other libraries, the SL8500 does not have CAPs defined for each LSM ID in an SL8500 library. The CAPs on an SL8500 contain LSM 1 in their CAP IDs. There are no CAPs in an SL8500 with LSM IDs 0, 2, or 3. Partitioning complicates this problem because LSM 1 (the LSM ID in the SL8500 CAP ID) may not be assigned to your partition. (Remember, the CAPs are still available to all partitions as a shared resource.)

Some ACSLS clients do not query ACSLS to identify which CAPs exist and are available before selecting a CAP for an enter, eject, or audit. They may specify *cap_ids* that do not exist or CAPs that are not online. For example, some ACSAPI clients assume CAPs exist for all LSM IDs. They may automatically specify a CAP with the same LSM ID as the location of cartridges or drives that they manage. Enters, ejects, or audits that specify non-existent CAP IDs will fail.

You need to use the ACSLS `cmd_proc` to:

• enter and eject cartridges for clients that specify non-existent CAP IDs.

• run audits for any ACSs and partitions used by these clients.

After entering, ejecting, or auditing, you need to re-synchronize the client application's database with the ACSLS database.

■ Minimizing Elevator and PTP Activity

There are several things you can do to minimize elevator and PTP activity, such as:

Mounting cartridges	Whenever possible when mounting a tape, use cartridges and tape drives that are in the same LSM.
	LSM refers to a single rail within the SL8500 library. Each SL8500 contains 4 LSMs.
Using float	Take advantage of the ACSLS "float" option (enabled by default by ACSLS) by maintaining some free cells within each LSM. Cartridge float is a feature that allows ACSLS to place a dismounted tape cartridge in an empty slot in the same LSM or a closer LSM to the tape drive if the tape originally came from a different LSM using a pass-thru operation.
	When cartridges are dismounted, ACSLS tries to avoid elevator (pass-thru) activity among LSMs by assigning a new home cell whenever the cartridge's old home cell is in a different LSM. ACSLS attempts to put the cartridge away:
	• in the same LSM as the tape drive from which it was dismounted
	• or to the closest LSM (with free storage cells) to the drive

Entering cartridges   Enter cartridges into an LSM that has compatible tape drives for the media being entered.

Example: You have only LTO drives on LSMs 2 and 3, and you want the LTO cartridges to be located in these LSMs. When entering these cartridges, you should place them in the CAP magazines adjacent to LSMs 2 and 3. ACSLS then makes every effort to put the cartridges in the LSM that is adjacent to that CAP magazine.

Scratch cartridges   Make sure that scratch cartridges are available in sufficient quantity in each LSM where they will be used. For an SL8500, this means having scratch cartridges available on each rail (LSM) of the library.

Free cells   Make sure there are adequate free cells in each LSM.

■ Configuring Tape Drives to Support Your Workloads

How tape drives are configured in the SL8500 can minimize both elevator and PTP activity while supporting your tape workloads. Strategies to use in determining where tape drives are located in the SL8500 include:

- Cluster cartridges by workload, with enough drives to support the maximum drives needed for the workload. Separate the cartridges used by each workload on separate rails, and ensure the rail(s) dedicated to a workload has enough drives to meet the maximum concurrent mounts for the peak usage of the workload. Ensure that the rail has not only the tape cartridges for the workload, but also the scratch cartridges that will be needed.

- Allocate separate rail(s) to each major application workload. This is because some applications, such as Symantic NetBackup and Tivoli, can use their own media and drives.

- Clustering drives and media on a single rail works until the mounts per hour threshold is reached, all drives are in use, or there are too many active cartridges to fit on a rail. When the resources needed for a workload exceeds the capacity of a rail, spread the cartridges and drives over two or more rails.

- Cluster drives by type, placing drives that use different media types on separate rails (LSMs). For example, place T9840 drives on one rail and T10000 drives on a different rail.

- Configure your heavy tape applications so they will not exceed the performance limits of your library configuration.

- Configure the SL8500 with 8 HandBots (two HandBots per rail) to provide redundancy. This allows you to always access the cartridges and drives that support a workload.

■ Managing Cartridge Locations

How cartridges are originally entered in the library or their status in the library can have an affect on ACSLS performance. Considerations are:

Entering cartridges	*Recommendation*: Enter cartridges through the CAP.
	When manually placing cartridges in the library with the front access door open, *library operations cease* and ACSLS must perform a *full audit* to update the library database to match the actual contents of the library.
	To maximize performance: Enter cartridges through the cartridge access port (CAP).
	During an enter, *the library stays online*, *mounts can continue*, and the library management software always tries to move the cartridge to an LSM adjacent to the CAP magazine—minimizing pass-thru activity.
	If this is not possible, the library controller moves the cartridge through the elevator to another LSM—which requires additional movement between two HandBots and the elevator.
Clustering cartridges	Cluster cartridges by workload on separate rails with enough tape drives to support the maximum activity—peak usage—for that workload.
Using float	*Recommendation*: When *float* is on, ACSLS selects a new home cell for a cartridge that is in an LSM as close to the drive as possible on a dismount. This option automatically clusters cartridges by the drives for the workload.
	Make sure each LSM contains enough free cells to allow selection of a new home cell in that LSM.
Supplying scratch cartridges	Make sure each rail has the correct amount and type of data cartridges *plus* enough scratch cartridges to support the workload.

▪ Finding Missing Cartridges

If a cartridge is out of place or unaccounted for by ACSLS:

1. Perform a physical audit of the SL8500 using the SLConsole.

   The physical audit of the SL8500 is performed as a background task in between handling mount and other library operation requests.

   > **Warning:** If the SL8500 contents are out of sync with ACSLS due to manual operations such as loading cartridges directly, it is not advisable to attempt continued operations.

   > If you want to manually add tapes, adding them to a particular LSM within the SL8500 is a better approach. Adding tapes to a particular LSM and auditing only the affected LSM is a quicker and more reliable solution.

   > To do this, vary the affected LSM to a diagnostic state to ACSLS while the audit is in process. After the SL8500 library audit is performed vary the LSM online to ACSLS.

2. Run an ACSLS audit to update the ACSLS database to match the actual inventory of library cartridges.

▪ Varying the SL8500 Offline

You should vary SL8500 components offline to ACSLS before they are powered off, if they are inoperative, and before you open an SL8500 access door. This notifies ACSLS that they are unavailable. Once they are available, vary them back online.

Use ACSLS to vary SL8500 components offline, not SLConsole

You should vary SL8500 components (ACSs, LSMs, and CAPs) offline to ACSLS, not the SLConsole.

ACSLS allows outstanding requests to complete before taking components offline, unless it is a vary offline force. The SLConsole has no knowledge of outstanding requests to ACSLS.

Varying components offline using SLConsole may cause requests in progress to fail.

When to vary SL8500 components offline to ACSLS

Before opening the access door

Before opening the SL8500 access door vary the ACS or all four LSMs offline.

- For a standalone SL8500, vary the ACS offline:

      vary acs *acs_id* offline

- For a SL8500 connected through PTPs, vary all four LSMs (in the SL8500 whose access door will be opened) offline using the following command four times (once for each of the four LSMs):

      vary lsm *lsm_id* offline

Note: If any CAPs in the SL8500 are in automatic mode, you must:

1. Set them to manual mode before opening the access door.

2. Set them back to automatic mode after you close the access door and the SL8500 comes back online.

If a CAP is inoperative

If the CAP is inoperative, vary it offline:

      vary cap *cap_id* offline

When closing the Service Safety Door

Whenever replacing hardware requires using the Service Safety Door, it is advisable to keep that Service Safety Door closed for the minimum amount of time possible. This is because the Service Safety Door blocks other hardware components (elevators, CAPs, and cells) to which access may be required for completing specific requests.

- Before closing the Service Safety Door on either the left or right side of the SL8500, vary the elevator on that side offline through the SLConsole.

  After the Service Safety Door is opened, vary the elevator on that side back online through the SLConsole.

- When the Service Safety Door is closed on the right side, it will block access to the CAP.

  - Before closing the Service Safety Door on the right side of the SL8500, vary the CAP offline through ACSLS.

  - After the Service Safety Door is opened, vary the CAP online through ACSLS.

Note: When the SL8500 Service Safety Door is closed to separate a service bay from the rest of the library, the CSE can open the access door on that side without taking the LSM or ACS offline.

When using the Service Safety Door do not use these ACSLS commands and utilities

There are some ACSLS commands and utilities that should not be in progress or initiated when the Service Safety Door is being used. These commands are:

When the Service Safety Door is closed on either side, do not use these utilities:

- `acsss_config`

- `config` (`config drives` is OK)

When the Service Safety Door is closed on the right (CAP) side, do not use these commands:

- `enter`

- `eject`

- `set cap mode auto <cap_id>`

When the Service Safety Door is closed on the right (CAP) side, the following commands can be used, but special considerations apply:

- `audit`

  The audit command can be used. However, if there is a need to eject cartridges as a result of the audit (e.g. because the audit encounters duplicates or unreadable labels), the audit will complete and update the ACSLS database, but the cartridges will **not** be ejected.

- `vary acs` and `vary lsm`

  These commands will succeed, but messages will be displayed on `cmd_proc` and the event log reporting CAP failures and inoperative CAPs.

■ Using the Dynamic Configuration (config) utility

The dynamic configuration (config) utility allows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration changes are recorded in the `acsss_config.log` file.

The following dynamic configuration utilities are supported:

- `config acs`
- `config drives`
- `config lsm`
- `config ports`

Using the `config` utility provides the following benefits:

- ACSLS can continue running, allowing you to perform mount requests to unaffected library components.

- Allows you to reconfigure specified library components while all other configuration information remains unchanged. For example, when specifying:

  - An ACS, the configurations of other ACSs are not affected.

  - An LSM, the configurations of other LSMs are not affected.

  - A drive panel, the drives on a panel, mounts and dismounts to all existing drives are not affected.

■ Using ACSLS HA

ACSLS High Availability (HA) is a software/solution offered by Professional Services. This solution is available for the Solaris SPARC 9 and 10 platforms.

In environments where there is only one ACS, it is sometimes desirable to fail over highly available servers in the event that communication is lost to the library. The ACSLS HA agent contains a variable that causes different behaviors based on these environmental considerations called "FAIL_OVER". This variable exists within the script located in `/opt/VRTSvcs/bin/STKLMU/monitor`, and can contain one of two values, "0" or "1".

- Setting the variable to "0" prevents failover

- Setting the variable to 1

- You can have the primary ACSLS server automatically fail over to the standby server in an attempt to resolve library communication failures.

  CAUTION: **Setting this variable to "1" is NOT recommended in environments with multiple libraries. If this is set in environments with multiple libraries, failing over will impact ALL libraries.**

- You can have the primary ACSLS server automatically fail over to the standby server in an attempt to resolve library communication failures.

Note: Please refer to the Readme for PTF830785 for a complete description of the FAIL_OVER variable

■ Auditing the SL8500 After Expansion

Expansion occurs when Storage Expansion Modules are added to the SL8500 to increase its capacity. When this happens the Customer Interface Module (CIM), which includes the CAP, must move out. As a result, newer and higher panel numbers are assigned to the three cell panels on the CIM.

When the cell panels on the CIM are assigned higher panel numbers, the addresses of all the cartridges on the CIM change. You must audit these panels so ACSLS can update its database with the new addresses of these cartridges.

Note: If you have to remove any cartridges to attach a new Storage Expansion Module, place them in an empty panel in the new Storage Expansion Module. Then audit this panel after auditing the three highest panels in the CIM. This guarantees audit will find these cartridges in their new address.

Example:

In the following diagram, the three cell panels in the Customer Expansion Module (# 4) are the three columns of cartridges on both the outside and inside walls on the drive side of the CEM. These are the only cell addresses in the CIM that HandBots can access when the service safety door is activated.

Figure 18. SL8500 Customer Expansion Module

L203_054

1. Cartridge Access Ports (2) Caps
2. Facade Operator Panel (Optional) Icey Pad
3. Customer Interface Module
4. Storage Expansion Module
5. Robotics Interface Module
6. PTP (Pass Through Port)
7. Drive Electronics Module
8. AC Power Supplies Electronic Control Module
9. DC Power Supplies
10. Tape Drives
11. Accessory Racks
12. Inner Wall
13. Service Door
14. Reserved Columns
   E= End Stop
   S= System Cell
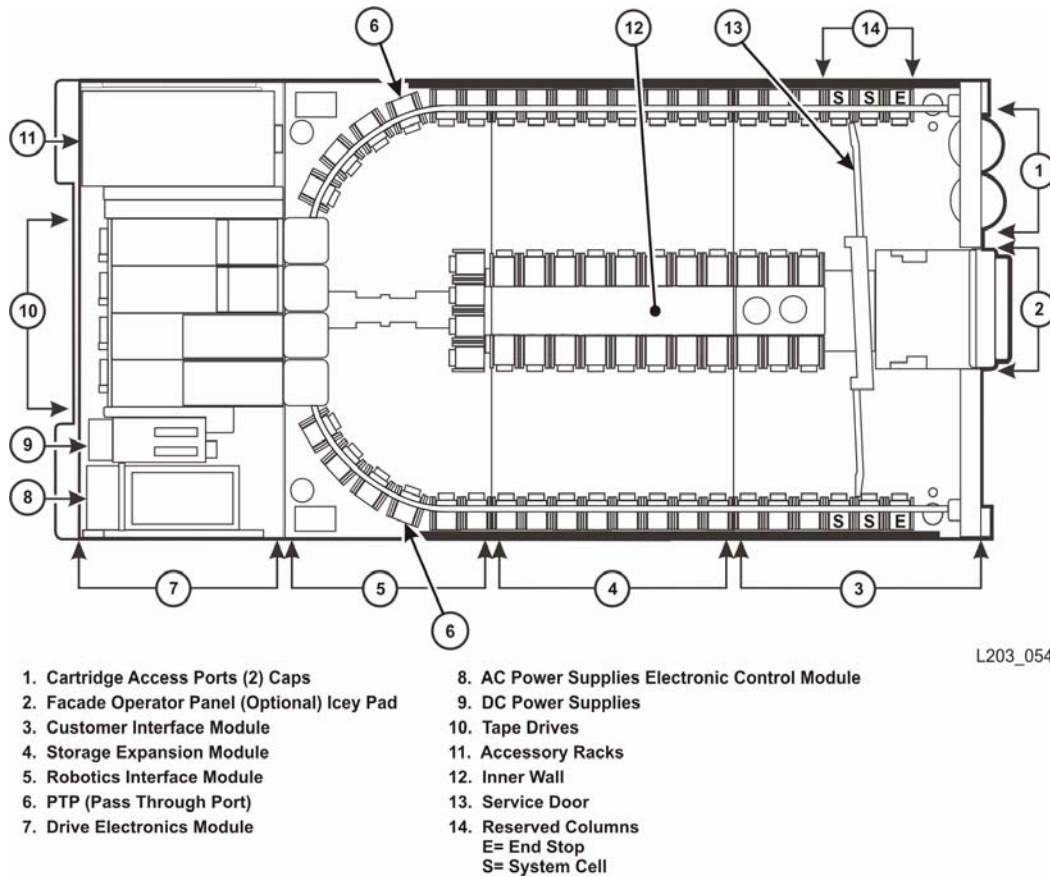
Auditing New Cartridge Addresses

When auditing the SL8500 after an expansion:

1. First, find and update cartridges whose addresses have changed by auditing the three highest panels in each LSM (rail).

    •   When audit finds a cartridge in a new address, it checks the old address where it was previously recorded in the ACSLS database to verify that this is not a duplicate cartridge.

    •   If a cartridge has changed addresses, ACSLS updates its database with the cartridge's new address.

        WARNING: If ACSLS first audits old cartridge addresses after an expansion, it will not find the cartridges and will delete them or mark them absent. If the cartridges are marked absent, they will be reactivated later when ACSLS audits their new address. However, if

> **the cartridge is marked as scratch status, it will lose that status. If the cartridges are deleted, information such as locks, ownership, cleaning cartridge usage, scratch status, and pool assignments are all lost.**

2. Last, audit the entire LSM.

   When you audit the entire LSM *last*, you do not lose any cartridge information.

Procedure for Auditing an Expanded SL8500

After an expansion has occurred

1. Vary the LSMs to a diagnostic state until the configuration is updated and cartridge addresses have been updated with audit.

2. Backup the ACSLS database.

3. Update the configuration in the ACSLS database using one of the following.

   - Dynamically:

     `config acs acs_id`

     or

     `config lsm lsm_id`   (for each LSM (rail) in the SL8500)

   - Statistically:

     a. Shutdown ACSLS: `kill.acsss`

     b. Update the ACSLS configuration: `acsss_config`

     c. Bring up ACSLS: `rc.acsss`

4. Determine the panel numbers for the three panels on the Customer Interface Module.

   If you placed cartridges that were removed to install the Storage Expansion Module(s) in a new panel, determine this panel number.

   display panel *acs,lsm,* *

   Where:

   acs is the ACS.

   lsm is one of the LSMs. All rails (LSMs) on the SL8500 will have the same number of panels

   * displays all panels in the LSM.

5. First, audit the three panels with the highest panel numbers first in each LSM (rail) in the expanded SL8500.

Then, audit the panel(s) where cartridges that were removed during the installation of the Storage Expansion Module(s) were placed.

6. Vary the LSMs online.

Auditing the panels, above, should have updated the addresses for all cartridges that moved.

7. Audit each entire LSM in the expanded SL8500.

8. Backup the ACSLS database.

■ Connecting SL8500s with Pass-Thru-Ports

You may need to audit SL8500(s) to update cartridge addresses when adding SL8500 libraries to an existing SL8500 library complex.

LSMs in the SL8500 complex are numbered from top down and right to left when viewed from the CAP end, as shown in the following figure.

Figure 19. Four connected SL8500 Libraries

Left	⇦		⇦		Right
Library 4		**Library 3**		**Library 2**		**Library 1**
LSM 12		LSM 8		LSM 4		LSM 0
LSM 13	P T P	LSM 9	P T P	LSM 5	P T P	LSM 1
LSM 14		LSM 10		LSM 6		LSM 2
LSM 15		LSM 11		LSM 7		LSM 3

Depending on how your site is set up, you may be adding a new SL8500 to the right or left of an existing SL8500. As a result:

• When a a new SL8500 is added on the left, the new LSMs have higher numbers, and existing LSMs are not re-numbered. The disruption is minimal.

• When a new SL8500 is added on the right, all existing LSMs are re-numbered. This causes the home cell addresses of all existing cartridges to change.

Before You Install SL8500 PTP Connections

1. Apply all relevant maintenance to ACSLS before installing SL8500 connections.

2. Enable absent cartridge support.

When absent cartridge support is enabled, ACSLS will not delete cartridges in an LSM that is deleted or when audit does not find them in their old address. These cartridges are marked absent, preserving all information about them, except for the scratch status. When these cartridges are later found by audit, they are reactivated.

- Absent cartridge support is enabled when the "Variable: ABSENT_VOLUME_RETENTION_PERIOD" on page 41 is **not** zero. The default value is 5 days.

- Use `acsss_config` (option 3). For procedures, refer to the "Variable: ABSENT_VOLUME_RETENTION_PERIOD" on page 41 under Option 3 - Set General Product Behavior Variables.

3. Update your configuration by one of the following methods.

- Dynamically:

      config acs *acs_id*

- Statically:

  a. Shutdown ACSLS: `kill.acsss`

  b. Update the ACSLS configuration: `acsss_config`

  c. Bring up ACSLS: `rc.acsss`

  ACSLS cannot be running when you run `acsss_config`.

  For more information, refer to "Configuring or Updating your Library Hardware" on page 46 or under the "Dynamic Configuration (config) utilities" on page 133.

Adding New SL8500s

When additional SL8500s are added to an existing SL8500 library complex, the new ACSLS configuration must be updated. If the addition of new SL8500s causes the LSMs in the existing SL8500s to be renumbered, the cartridge addresses in those LSMs must be updated.

The cartridge addresses should be updated without losing the other information associated with the cartridges. This includes cartridge information, such as scratch status, pool, locks, and ownership, as well as cleaning cartridge usage counts, and the date and time cartridges were entered.

LSMs in connected SL8500s are numbered from top to bottom and right to left (as viewed from the CAP end).

Figure 20. Four connected SL8500 Libraries

Left		Left ⇐		⇐		Right
Library 4		**Library 3**		**Library 2**		**Library 1**
LSM 12	P T P	LSM 8	P T P	LSM 4	P T P	LSM 0
LSM 13		LSM 9		LSM 5		LSM 1
LSM 14		LSM 10		LSM 6		LSM 2
LSM 15		LSM 11		LSM 7		LSM 3
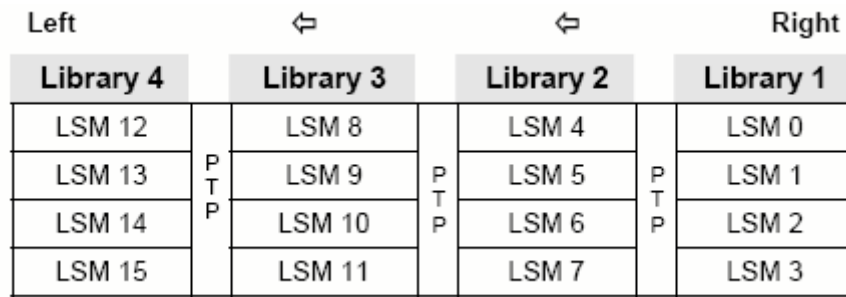
Adding a New SL8500 to the Left

In the case of the non-disruptive addition of a new SL8500 on the left (from the CAP end) of the original SL8500(s), you can use the host software (ACSLS) to dynamically configure the additional drives and libraries. You do not need to re-cycle ACSLS if Dynamic Configuration is installed. Mount requests continue as normal to the existing SL8500s during this phase.

If cartridges were placed in cells in the new SL8500, an ACSLS audit must be run on the LSMs of the new SL8500 to add these cartridges to the ACSLS database.

The LSMs in the pre-existing SL8500s can be online during the audit.

Dynamically Configuring the New ACSLS Configuration

1. Add new SL8500(s) to the library complex.

2. Backup ACSLS before you make the configuration change.

3. Update the ACSLS configuration dynamically:

    ```
    config acs acs_id
    ```

    You can also update the ACSLS configuration when ACSLS is down:

    ```
    acsss_config
    ```

4. Backup ACSLS after you make the configuration change.

    Note: If there are cartridges in the new SL8500, audit the LSMs (rails) in the new SL8500 to add these cartridges to the ACSLS database.

Adding a New SL8500 to the Right

If you wish to add a new SL8500 to the right, the existing LSMs will be renumbered, as shown in the following figures

Considerations When Adding an SL8500 to the Right

When new SL8500s are added to the right, all existing LSMs will be renumbered and cartridge addresses will change. The change to LSM numbers causes all existing cartridge addresses to change. If ACSLS attempts to mount a cartridge whose address changed, the mount will fail because ACSLS cannot find the cartridge.

- Quiesce mount activity until all cartridge addresses have been updated.

- Vary all LSMs in the ACS to diagnostic state to prevent mounts.

- Update the cartridge addresses by auditing the LSMs in the existing and new SL8500s in a specific sequence.

Dynamically Configuring the New ACSLS Configuration

To a update the ACSLS Configuration Dynamically after adding a new SL8500(s):

1. Vary existing LSMs in diagnostic state (`vary lsm_id diag`).

   WARNING: **These LSMs must remain in diagnostic state until they have been audited. Otherwise these problems will occur:**

   - Mounts will fail because cartridges cannot be found in their last-known addresses.

   - Until the audits update the ACSLS map of the empty cells in the renumbered LSMs:

     - Enters of new cartridges will collide with existing cartridges.

     - Movements of cartridges to existing (renumbered) LSMs will collide with cartridges already in the cells,

2. Add the new SL8500(s) to the library complex.

3. Backup ACSLS (before making any configuration changes).

4. Update the ACSLS configuration dynamically using:

   ```
   config acs acs_id
   ```

   Note: As an alternative, you can update the ACSLS configurations while ACSLS is down. To do this:

   a. Bring ACSLS down using the `kill.acsss` command.

   b. Change the configuration with `acsss_config`.

    c. Bring ACSLS back up with `rc.acsss.`

> **Note:** New LSMs are added when in an online state. Vary these LSMs to a diagnostic state until they have been audited.

5. Backup ACSLS (after the configuration change).

6. Audit the library to update cartridge addresses (since they were renumbered).

    To do this, use the following the following sequence:

    a. First, audit each LSM in the pre-existing SL8500s that were renumbered.

        - **Audit the pre-existing SL8500s in a careful sequence to avoid losing cartridges. Audit each LSM (rail) in sequence from the highest numbered LSM to the lowest numbered LSM ID.** The audit will find all of the cartridges in their (renumbered) LSM addresses.

        - **For ACSLS, you must audit each LSM separately, and allow the audit of one LSM to finish before auditing the next LSM.** Audit will check the cartridges' former home cell addresses (in the LSM to the right), and then update the cartridges' addresses. The audits for the LSMs in the left most SL8500 will take a long time. The audits in all of the other SL8500s will be faster.

          The reason the audits of the left most LSMs take longer is that they will trigger a cascade of Cartridge Recovery requests. When audit finds a cartridge that has a different home cell address, audit checks the cell address recorded in the database. If that cell has a different cartridge, Cartridge Recovery investigates that cartridge, etc. Because all of the LSM addresses have changed, these recoveries will cascade across the SL8500 library complex.

        - Keep the LSMs in diagnostic state until they have been audited. After they have been audited, you can vary them online. Audit has updated the addresses of cartridges in this LSM, so you can resume automated mount activity with the audited LSM's cartridges.

    b. Last, audit the LSMs in the newly added SL8500 (that have the lowest LSM IDs).

        After these LSMs have been audited, they can be varied online and cartridges in them can be mounted on tape drives, as shown in the following figures.

Figure 21. Original Configuration Before an SL8500 is added to the Right

SL8500		SL8500		SL8500
LSM 8		LSM 4		LSM 0
LSM 9		LSM 5		LSM 1
LSM 10		LSM 6		LSM 2
LSM 11		LSM 7		LSM 3

Figure 22. Correct Audit Order (shown in bold) and New Configuration

First Four Audits		Second Four Audits		Third Four Audits		Last Four Audits
Now LSM 12 **(Audit # 4)**		Now LSM 8 **(Audit # 8)**		Now LSM 4 **(Audit # 12)**		New LSM 0 **(Audit # 16)**
Now LSM 13 **(Audit # 3)**	P T P	Now LSM 9 **(Audit # 7)**	P T P	Now LSM 5 **(Audit # 11)**	P T P	New LSM 1 **(Audit # 15)**
Now LSM 14 **(Audit # 2)**		Now LSM 10 **(Audit # 6)**		Now LSM 6 **(Audit # 10)**		New LSM 2 **(Audit # 14)**
Now LSM 15 **(Audit # 1)**		Now LSM 11 **(Audit # 5)**		Now LSM 7 **(Audit # 9)**		New LSM 3 **(Audit # 13)**

7. Backup ACSLS after the audits have completed.

Procedures for Merging ACSs

SL8500 PTP supports the merging of multiple, separate SL8500s into a single ACS. To minimize the loss of information about cartridges and operational downtime, you should follow these recommended procedures.

Note: When the ACSLS configuration is updated, global cartridge address changes **are not** made.

For ACSLS, two scenarios are described below. In the first, the ACSs to be merged are numbered from right to left (as you face them from the CAP side). In the second, they are numbered from left to right. It is assumed that the resulting ACS will use the lower/lowest ACS ID.

Merging ACSs Numbered from Right to Left

In the following scenario, the ACSs to be merged are numbered from right to left (as you face them from the CAP side).

Figure 23. Existing Configuration – Three Separate ACSs

ACS 2		ACS 1		ACS 0
LSM 2,0		LSM 1,0		LSM 0,0
LSM 2,1		LSM 1,1		LSM 0,1
LSM 2,2		LSM 1,2		LSM 0,2
LSM 2,3		LSM 1,3		LSM 0,3

Figure 24. Desired Configuration – Single ACS

		ACS 0		
LSM 0,8		LSM 0,4		LSM 0,0
LSM 0,9	**P T P**	LSM 0,5	**P T P**	LSM 0,1
LSM 0,10		LSM 0,6		LSM 0,2
LSM 0,11		LSM 0,7		LSM 0,3

Procedure for Merging ACSs Numbered from Right to Left

1. Vary all ACSs, except the right most ACS being merged, offline.

   This prevents mounts and dismounts while cartridge addresses are being updated.

2. Stop ACSLS.

3. Backup ACSLS before making any configuration changes.

4. Update the ACSLS configuration (while ACSLS is down) using `acsss_config`

5. Bring up ACSLS.

6. Vary the new LSMs (that were added to the ACS) to a diagnostic state.

   These LSMs were added in while in online state.

7. Backup ACSLS after the configuration change.

8. Audit the LSMs that were added to the ACS

The order in which they are audited does not matter. The entire ACS or all LSMs can be audited at once.

The cartridges will be re-activated.

9. Backup ACSLS after the audits are complete.

10. Vary the new LSMs online and resume normal automated processing.

Merging ACSs Numbered from Left to Right

In the following scenario the ACSs to be merged are numbered from left to right.

Figure 25. Existing Configuration - Three Separate ACSs

ACS 0		ACS 1		ACS 2
LSM 0,0		LSM 1,0		LSM 2,0
LSM 0,1		LSM 1,1		LSM 2,1
LSM 0,2		LSM 1,2		LSM 2,2
LSM 0,3		LSM 1,3		LSM 2,3

Figure 26. Desired Configuration - Single ACS

ACS 0				
LSM 0,8		LSM 0,4		LSM 0,0
LSM 0,9	P T P	LSM 0,5	P T P	LSM 0,1
LSM 0,10		LSM 0,6		LSM 0,2
LSM 0,11		LSM 0,7		LSM 0,3

Procedure for Merging ACSs Numbered from Left to Right

1. Vary all ACSs offline.

    This prevents mounts and dismounts while cartridge addresses are being updated.

2. Stop ACSLS.

3. Backup ACSLS before making any configuration changes.

4. Update the ACSLS configuration (while ACSLS is down) using `acsss_config.`

5. Bring up ACSLS.

6. Vary the new LSMs (that were added to the ACS) to a diagnostic state.

(These LSMs were added in while in online state).

7. Backup ACSLS after the configuration change.

8. Vary the new LSMs that were added to the left to a diagnostic state.

   You do not want to perform mounts and dismounts until you have audited these LSMs.

9. Audit the newly configured LSMs.

   The order in which they are audited **does** matter. Audit in this order:

   a. **First, audit the LSMs in the SL8500(s) that retains the same ACS number first.**

      - You want audit to find these cartridges in their new LSM addresses before you audit the LSMs that now are assigned to these cartridges' old addresses.

      - The LSM addresses will be updated for all cartridges in these SL8500(s).

      - **For ACSLS, you must audit each LSM separately, and allow the audit of one LSM to finish before auditing the next LSM.**

      - Keep the LSMs in diagnostic state until they have been audited.

        After they have been audited, you can vary them online. Audit has updated the cartridge's addresses, so you can resume automated mount activity with the audited LSM's cartridges.

   b. Last, audit the SL8500s that were merged into the lowest ACS. The cartridges in these SL8500s will be re-activated.

      - All of these LSMs can be audited at the same time.

      - The order in which these LSMs are audited does not matter.

10. Backup ACSLS (after the audits are complete)

11. Vary all LSMs online and resume normal automated processing.

Removing PTPs and Splitting ACSs

You may decide to remove the PTP mechanisms that connect two SL8500s and split a single ACS into two separate ACSs. This configuration change is the reverse of merging two ACSs into a single ACS.

It is much easier to add the new ACS with the SL8500(s) on the left hand side of the split. This assigns the higher numbered LSMs to the new ACS and avoids renumbering the LSMs that remain in the existing ACS.

Splitting ACSs, with the new ACS Created from SL8500(s) on the Left - Possible Scenarios

Figure 27. Existing Configuration – One ACS

ACS 0

LSM 0,12	P T P	LSM 0,8	P T P	LSM 0,4	P T P	LSM 0,0
LSM 0,13		LSM 0,9		LSM 0,5		LSM 0,1
LSM 0,14		LSM 0,10		LSM 0,6		LSM 0,2
LSM 0,15		LSM 0,11		LSM 0,7		LSM 0,3

Figure 28. Desired Configuration - Two ACSs

ACS 1   **ACS 0**

LSM 1,4	P T P	LSM 1,0	LSM 0,4	P T P	LSM 0,0
LSM 1,5		LSM 1,1	LSM 0,5		LSM 0,1
LSM 1,6		LSM 1,2	LSM 0,6		LSM 0,2
LSM 1,7		LSM 1,3	LSM 0,7		LSM 0,3

ACSLS Procedure for Splitting an ACS

With Dynamic Configuration, ACSLS can add ACSs while it is running. However, Dynamic Configuration does not make global cartridge address changes.

1. Backup ACSLS before the configuration change.

2. Vary the LSMs that will be moved to the new ACS offline.

    This prevents mounts and dismounts while cartridge addresses are being updated.

    The LSMs that are remaining in the existing ACS can remain online.

    Mounts and dismounts in these LSMs can continue.

3. Remove the four PTP mechanisms that connect the SL8500s that you want to separate.

4. Reconfigure the existing ACS once the pass-thru ports have been removed.

    This will remove the LSMs that are moving to the new ACS.

        config acs *acs_id*

5. Add the new ACS.

```
config acs acs_id new
```

Alternatively, you can update the ACSLS configurations when ACSLS is down:

```
acsss_config
```

New LSMs are added in an online state.

6. Vary these LSMs in a diagnostic state until they have been audited.

7. Vary the new ACS to diagnostic state.

8. Backup ACSLS after the configuration change.

9. Audit the LSMs in the new ACS.

   You do not want to perform mounts and dismounts until you have audited these LSMs.

   The order in which they are audited does not matter. They can all be audited at once.

   The cartridges will be re-activated.

10. Backup ACSLS after the audits are complete.

11. Vary the new LSMs online and resume normal automated processing.

    LSMs can be varied online as soon as the audit of that LSM completes.

Adding the New ACS on the Right Hand side of the Split

If the new ACS is added on the right hand of the split, all cartridge addresses will be re-mapped.

As this is not recommended, a detailed procedure is not provided.

However, some considerations are:

- When the LSMs in the right hand SL8500(s) are removed from the existing ACS, the cartridges in these LSMs will be marked absent (if absent cartridge retention is active). They will be reactivated when the new ACS is audited.

- The LSMs in pre-existing ACS must be audited one LSM at a time to update the addresses of the cartridges in these LSMs. Audit the highest numbered LSM first, followed by the next lower LSM. The audit of one LSM must finish before the next audit is started.

■ ACSLS Dual TCP/IP Support

Dual TCP/IP is an option that can be purchased for the SL8500 and SL3000 libraries (herein known as the library). It provides two TCP/IP connections to the library. However, you can continue to use the library with only one of the two connections operational.

The purpose of dual TCP/IP is to automatically recognize and avoid a failing communication path. Since this is automated, there is no need for you to manually switch from an inoperative connection.

To use dual TCP/IP support on the library, the routing tables on both the ACSLS server and the library must be managed using the "route" command. This forces a route to the defined network interfaces on the library which in essence, creates a one to one relationship between interfaces. The Customer Systems Administrator (CSA) changes the routing tables on the ACS server and the Customer Systems Engineer (CSE) updates the routing tables on the library. For further information on the UNIX "route" command, please refer to the manpages on your ACSLS server.

Requirements

- Please coordinate with both your system administrator and network administrator to understand your current network environment and to identify all necessary IP addresses in advance.

- Please coordinate with your system administrator to either configure your network interface or to validate that it is configured properly.

- Verify you have the following:

  - ACSLS 7.3 for Solaris 10 Update 4 for Solaris/SPARC and Solaris/x86

  - ACSLS 7.3 for AIX 5.3 TL6

Configuration

It is recommended that ACSLS keep two connections to the library open since ACSLS uses all active connections. If one connection is inoperative ACSLS uses the remaining operative connection, while continuing to try to re-establish communication on the failing connection.

The preferred configuration for dual TCP/IP implementations would be two network interfaces on two separate subnets for the ACSLS server as shown in Scenario 1. This provides maximum throughput and minimum resource contention with regard to network communications while adding a second physical connection improving reliability.

To configure two TCP/IP connections to a single library use the `acsss_config` utility or Dynamic Configuration (config). You must enter the number (2) of connections there are to the library and the IP addresses of the network devices.

The following scenarios provide examples for configuring the ACSLS server. For instructions on configuring the library dual TCP/IP feature, please refer to appropriate library System Dual TCP/IP Feature document.

The following scenarios use private subnet IP addresses and will not be the same in your environment. These scenarios assume that your network devices have been configured and are functioning properly.

The following scenarios show an SL8500. For this release, the SL3000 does not support ACSLS 7.3 HA.

Scenario 1

Scenario 1 is the preferred configuration for the dual TCP/IP feature.

In this configuration, the ACSLS server contains two network interfaces that reside on two separate subnets. The SL8500 also contains two network interfaces that reside on the same two subnets as the ACSLS server.
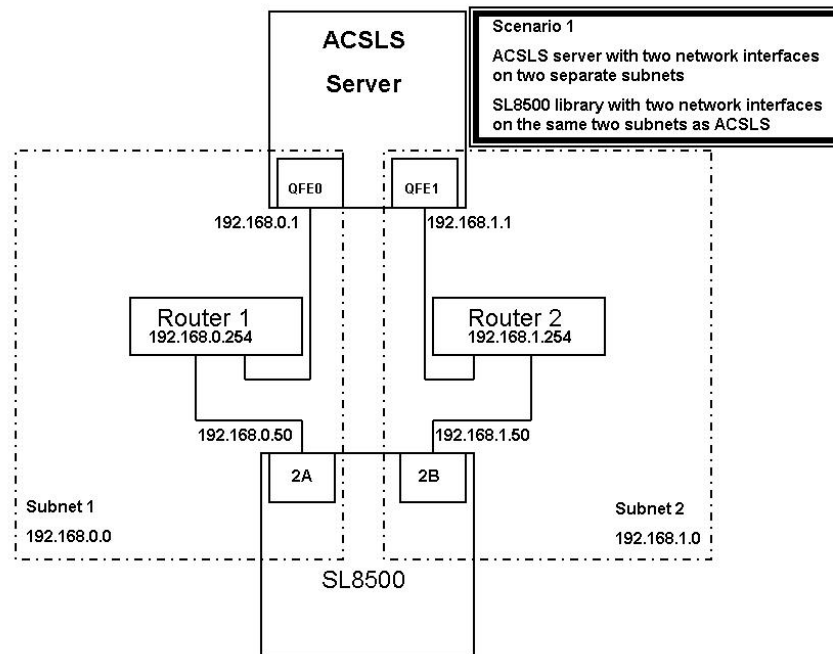
In this scenario, the SL8500 uses a one to one relationship with the network interfaces on the ACSLS server in which the:

- qfe0 interface on the ACSLS server only communicates with the 2A interface on the SL8500

- qfe1 interface on ACSLS only communicates with 2B on the SL8500.

Using the UNIX "route" commands, you force this relationship. To do this:

- For Solaris: as user `root`, type the following commands:

  #route add 192.168.0.50 -ifp qfe0 192.168.0.254

  #route add 192.168.1.50 -ifp qfe1 192.168.1.254

- For AIX: as user `root`, type the following commands:

  #route add 192.168.0.50 -if qfe0 192.168.0.254

  #route add 192.168.1.50 -if qfe1 192.168.1.254

The first route command routes any communication with 192.168.0.50 to go through qfe0 on the ACSLS server and then go through Router 1.

The second command routes any communication with 192.168.1.50 to go through qfe1 on the ACSLS server and then go through router 2.

You can validate that the routes are in the routing table by typing:

  # intestate –r

Routing Table: IPv4

```
Destination             Gateway            Flags Ref   Use
Interface

------------------- ------------------- ----- ----- ------ --
-------

192.168.0.50            192.168.0.254       UGH   1     0     qfe0
192.168.1.50            192.168.1.254       UGH   1     0     qfe1
192.168.0.0             192.168.0.1         U     1     7     qfe0
192.168.1.0             192.168.1.1         U     1     0     qfe1
BASE-ADDRESS.MCAST.NET 192.168.0.1          U     1     0     qfe0
default                 192.168.0.254       UG    1     33
localhost               localhost           UH    4     77    lo0
```

You can see the first two entries are the ones that were just added. All communication with 192.168.0.50 will go through qfe0 and communication with 192.168.1.50 will go through qfe1.

Remember: You must also configure the SL8500 routing tables according to the instructions in the *StreamLine SL8500 Modular Library System Dual TCP/IP Feature* document.

Scenario 2

Scenario 2 also shows two interfaces on the ACSLS server on two separate subnets but communication to the SL8500 will pass through the public network. Even with this difference, the commands stay the same.

Using the UNIX "route" commands, you force this relationship. To do this:

- For Solaris: as usr `root`, type the following commands:

  #route add 192.168.2.50 -ifp qfe0 192.168.0.254

  #route add 192.168.3.50 -ifp qfe1 192.168.1.254

- For AIX: as user `root`, type the following commands:

  #route add 192.168.0.50 -if qfe0 192.168.0.254

  #route add 192.168.1.50 -if qfe1 192.168.1.254

The default routes for the ACSLS remain the same. The routes within the subnets will know how to route communication to the SL8500 through the public LAN and you are still forcing the one to one relationship with the interfaces. Again, this is seen using the following command:

    # netstat –r

```
Routing Table: IPv4

Destination          Gateway             Flags  Ref   Use
Interface

-------------------- -------------------- ----- ----- ------ --
-------

192.168.0.50         192.168.0.254        UGH   1     0      qfe0
192.168.1.50         192.168.1.254        UGH   1     0      qfe1
192.168.0.0          192.168.0.1          U     1     7      qfe0
192.168.1.0          192.168.1.1          U     1     0      qfe1
BASE-ADDRESS.MCAST.NET 192.168.0.1        U     1     0      qfe0
default              192.168.0.254        UG    1     33
localhost            localhost            UH    4     77     lo0
```

Remember: You must also configure the SL8500 routing tables according to the instructions in the *StreamLine SL8500 Modular Library System Dual TCP/ IP Feature* document.

Scenario 3

Scenario 3 **is not the preferred method** but will work. You can employ the dual TCP/IP feature on the SL8500 without using two network interfaces on the ACSLS server. This reduces reliability because if the ACSLS server network interface fails, communication with the library fails. If you do intend to implement this scenario, there Is no need for special configuration on the ACSLS server. The default route is already established for normal network configuration, and it would be redundant to add specialized routes using the same default route. It is worth noting here that you still need to configure the SL8500 according to the instructions in the *StreamLine SL8500 Modular Library System Dual TCP/IP Feature* document.

```
                                    ┌──────────────────────┐
        ACSLS                       │ Scenario 3:          │
        Server                      │                      │
                                    │ ACSLS server with one│
                                    │ network interfaces on a│
                                    │ separate subnet.     │
          QFE0                      │ SL8500 library with two│
                                    │ network interfaces on two│
   192.168.0.1                      │ subnets separate from the│
                                    │ ACSLS server.        │
                                    └──────────────────────┘
        Router 1          Subnet 1
        192.168.0.254     192.168.0.0

  Router 2                         Router 3
  192.168.1.254                    192.168.2.254
       192.168.0.50          192.168.2.50
          2A        2B

           SL8500

 Subnet 2                           Subnet 3
 192.168.1.0                        192.168.2.0
```
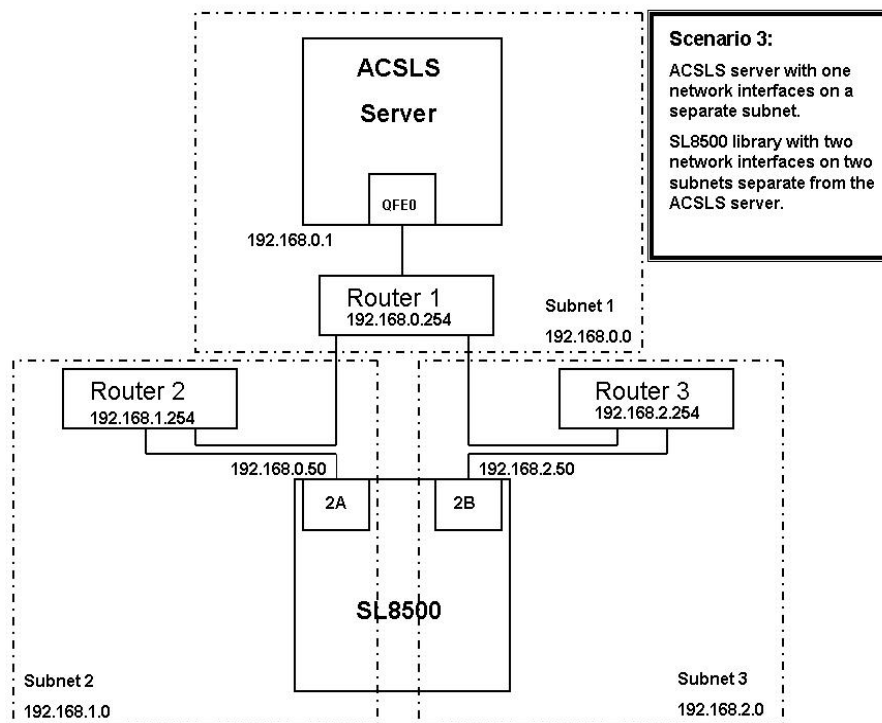
Scenario 4

Scenario 4 also shows the ACSLS server using a single network interface while the SL8500 has two interfaces on two different subnets. This also **is not a preferred method** since there is no redundant interface on the ACSLS server.

As in Scenario 3, there is no need for special configuration on the ACSLS server. Routing is already handled with a normal network configuration but the SL8500 still needs to be configured.

Remember: You must also configure the SL8500 routing tables according to the instructions in the *StreamLine SL8500 Modular Library System Dual TCP/IP Feature* document.
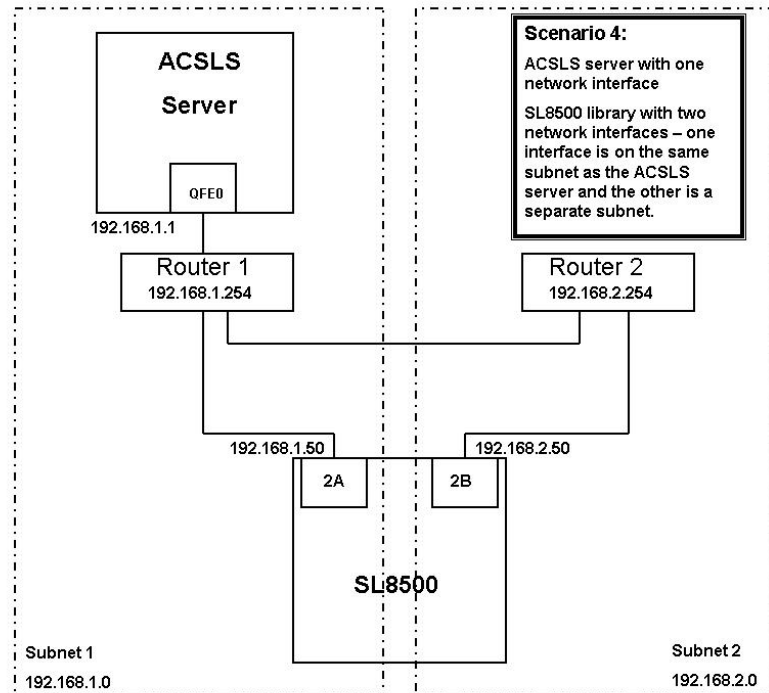
Scenario 5

Scenario 5 shows Highly Available ACSLS (ACSLS HA) in a dual TCP/IP environment.
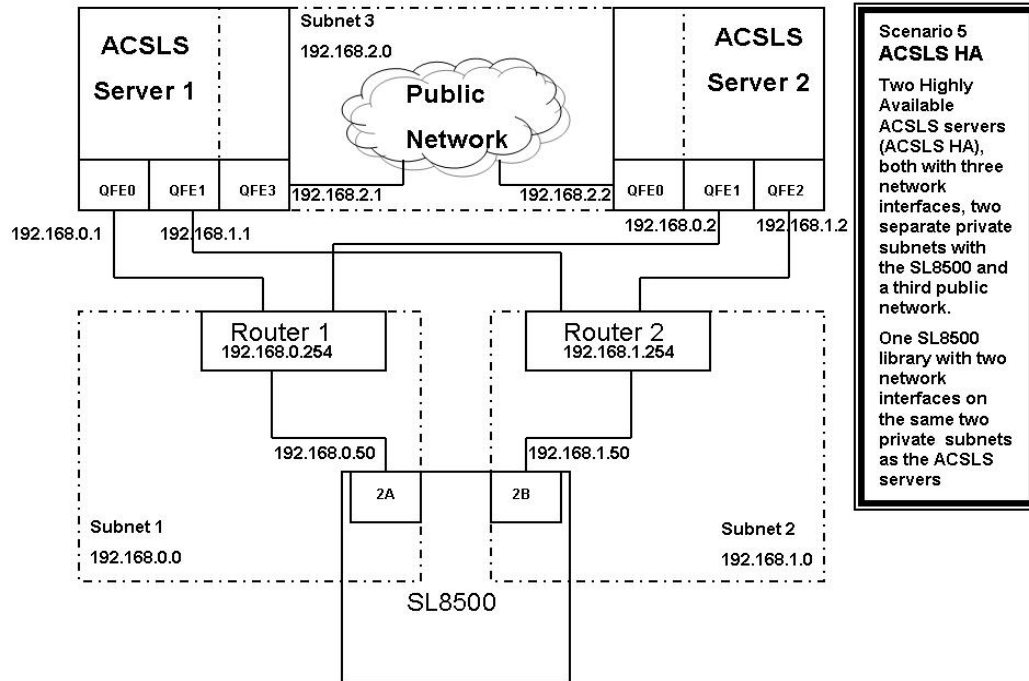
The big difference with this scenario is that ACSLS HA uses two different servers with each using different network interfaces. This means that custom route entries must be added to both ACSLS servers.

For the Solaris user:

- On ACSLS server 1, you would type:

    # route add 192.168.0.50 –ifp qfe0 192.168.0.254

    # route add 192.168.1.50 –ifp qfe1 192.168.1.254

- On ACSLS Server 2, you would type:

    # route add 192.168.0.50 –ifp qfe1 192.168.0.254

    # route add 192.168.1.50 –ifp qfe2 192.168.1.254

You must add the IP addresses for both servers to the SL8500 configuration. Please refer to the *StreamLine SL8500 Modular Library System Dual TCP/IP Feature* document.

It is important that you separate the SL8500 network interfaces over two different subnets when on ACSLS HA. The purpose of a Highly Available environment is to build in redundancy and eliminate single points of failure.

Remember: You must also configure the SL8500 routing tables according to the instructions in the StreamLine SL8500 Modular Library System Dual TCP/IP Feature document.

Retaining Customized Routing Table Entries after a Reboot

Any customized routing table entries will be lost after a system re-boot. This is the nature of system routing tables and is expected behavior.

In order to support the Dual TCP/IP feature on the SL8500, it is necessary to add custom entries to the routing tables on the ACSLS server. When the ACSLS server is re-booted, all routing table entries are flushed and any necessary routes to the SL8500 are removed. Since this is the nature of the operating system, there are a couple of different ways to handle this situation:

Creating Scripts

You can create scripts to add custom routes to be initialized at boot time. See for procedures.

These scripts can then be placed in the *rc* directory structure for automatic execution at boot time. Please refer to your system documentation for details on the best way to implement this.

Use the ACSLS startup scripts to add your custom routing entries at boot up time. The startup scripts check for a file that contains customized route table entries. Any entries found will be added to the routing table automatically using the UNIX route command.

For standalone ACSLS installations this is a desirable method to maintain route entries that are necessary for SL8500 support.

Important: This solution **will not work** if the ACSLS installation is a Highly Available ACSLS (ACSLS/HA) environment.

In this case, you need to use the first method to maintain routing tables.

ACSLS HA handles system initialization differently than a standalone ACSLS server. ACSLS HA relies on Veritas Cluster Server to manage its clustered resources which means ACSLS cannot be automatically started by way of the system RC mechanism at boot time.   This is handled strictly by the Veritas agents, and the S87ACSLS startup scripts are never used. You need to add a script with the appropriate "route add" commands and locate it within the /etc/

rc2.d directory structure. It is highly recommended that any customers with an ACSLS HA environment engage Sun Professional Services – preferably the same PS consultant that originally installed the ACSLS HA system.

Configuring ACSLS To Initialize Automatically At System Boot Time

To set up ACSLS to initialize automatically when the system is booted, execute **fix_rc.sh** This interactive utility determines the current behavior of your system, reports that to you, and then asks if you would like ACSLS to start automatically at boot time.

In the directory $ACS_HOME/install, as user **root**:

```
# ./fix_rc.sh
```

Your system is currently configured in such a manner that when it reboots, it will automatically restart the ACSLS.

Do you want your system configured so that when it reboots it automatically restarts the ACSLS? **(y or n):**

Simply answer **"y" or "n"** and the utility will complete the task.

```
Do you want your system configured so that when it
reboots it automatically restarts the ACSLS? (y or n):
y

fix_rc.sh 1864: Adding commands to automatically
restart ACSLS on reboot.
```

Note: If you choose not to automatically restart ACSCLS upon a reboot, then 'not classified' messages similar to the following are seen in the `acsss_event.log` after the server is rebooted:

```
2007-03-27 13:37:12 storage server[0]:
1328 N di_pri_get_status_code.c Unknown 188
di_pri_get_status_code:
 DBMS error.  Return code (28000) and message ("[unixODBC]")
not mappable to DI_STATUS

2007-03-27 13:37:12 storage server[0]:
1328 N di_pri_get_status_code.c Unknown 188
di_pri_get_status_code:
 DBMS error.  Return code (28000) and message
("[unixODBC]Communication closed during authentication;
```

Adding Custom Route To Be Initialized At Boot Time

To add custom routing entries:

1. cd to the following directory:

    $ACS_HOME/data/external/ custom_routing.

    This directory contains the template file *custom_routing_tables.tpl*.

2. Copy this file and change the file name to *custom_routing_tables*.

    # cp custom_routing_tables.tpl custom_routing_tables

3. Edit (vi) the *custom_routing_tables* file and add your entries.

    The file contains three fields.

    - The IP address for the SL8500

    - The name of the interface on the ACSLS server that you wish to establish the one-to-one relationship.

    - The IP address of the default route for your subnet.

4. Follow the instructions in the *custom_routing_tables* comment section for the format.

    WARNING: Make sure that there are no blank lines.

    When your server reboots, ACSLS is automatically initialized, and your custom routes are added to the routing table.

5. Verify all routes in the routing table with:

    # netstat -r

Please refer to your Unix man pages for complete documentation on both the "route" and "netstat" commands.

Removing routing commands

Use the route command to remove any special routing commands that have been added erroneously or are no longer needed to the earlier configuration.

Example: As user `root`, type the following commands:

route delete 192.168.0.50 192.168.0.254

This says to remove the route to 192.168.0.50 (the SL8500) using the default route of 192.168.0.254. The route is then removed.

ACSLS Support of the SL3000   C

The SL3000 offers the following benefits:

- Storage capacity from 200 to 4500 storage cells

- Performance from 1 to 56 tape drives

- Support of mixed media and a variety of tape drives

WARNING:

You must audit an SL3000 after it is configured or re-configured to ACSLS. The library reports the potential cell locations that are not accessible to ACSLS when ACSLS audits it. Inaccessible cell locations include:

- locations where CAPs, drives, and operator panels are installed.

- cell locations that the robots cannot access.

- cell locations that are not licensed.

- cell locations that are not in this partition.

■ ACSLS Support

ACSLS provides support for the following SL3000 features:

- New LSM and panel types and panel maps for the SL3000 Library

- Up to eight (8) partitions.

  The SL3000 can partition down to the drive and cell level. If cells are reassigned from one partition to a different partition, cartridges in those cells will be orphaned, and they will no longer be accessible by the partition that they were in before. To avoid this, before re-partitioning an SL3000, move cartridges to cells that will remain in your partition.

- New maximum numbers of CAPs (12), drives per panel (32), and cell rows (52) for the SL3000.

- Ability to dedicate a CAP to a partition within the SL3000.

  CAPS are dedicated to a partition using the SLConsole. The library reports dedicated CAPs to ACSLS.

  CAPs **dedicated** to a partition managed by ACSLS can be set to automatic mode.

- • SL3000 CAP Aliasing

  The SL3000 can have up to 12 CAPs, but some backup applications currently only support a maximum of 3 CAPs per LSM. CAP aliasing lets you alias CAPs 3-10 as CAPs 0, 1, or 2.

■ Before Configuring ACSLS for the SL3000

Before you configure ACSLS for the SL3000:

- • Connect the SL3000 to ACSLS

- • Verify that all the components of the SL3000 are operational.

  ACSLS builds its library configuration from the information reported by the library. If SL3000 components are not operational, the library information may not be reported to ACSLS, and the ACSLS configuration of the SL3000 could be incomplete.

Connecting the SL3000 to ACSLS

The SL3000 library uses TCP/IP protocol over an Ethernet physical interface to manage and communicate with the host and ACSLS. This interface enables ACSLS to connect to and communicate with the SL3000.

It is preferable to have the SL3000 library on a separate subnet or a controlled network to protect it from floods of ARP broadcasts.

Verifying that all SL3000 Components are Operational

To verify that all the components of the SL3000 are operational, logon to the StorageTek Library Console™ (SLConsole).

Refer to the *SL3000 User's Guide* for procedures.

Configuring SL3000 to ACSLS

Once the SL3000 components are operational, configure SL3000 to ACSLS. You can use either `acsss_config` (see "Configuring or Updating your Library Hardware" on page 46) or Dynamic Configuration (see "Dynamic Configuration (config) utilities" on page 133).

Auditing the SL3000

You must audit an SL3000 after it is configured or re-configured to ACSLS. The audit determines which cells are accessible to ACSLS and which potential cell locations are not installed, not accessible by the library, not licensed, or assigned to other partitions. Inaccessible cell locations include:

■ Understanding Internal Addressing

There are differences in the internal addresses of the SL3000 and other libraries supported by ACSLS and HSC.

- The SL3000 is *one's*-based (1) and uses negative numbers.
- Other libraries use a *zero*-based (0) with *no* negative numbers.
- The SL3000 uses *five* parameters: library, rail, column, side, and row.
- Other libraries use: ACS, LSM, panel, row, and column (HLI–PRC).

The modules of the library consists of walls, columns, and rows that house cartridges, tape drives, cartridge access ports, and robotic units.

You can use the StorageTek Library Console™ (SLConsole) to translate between the SL3000 internal address and the ACSLS panel, row, and column.

Refer to the *SL3000 User's Guide* for detailed information and procedures.

■ CAP Numbering

CAPs are numbered as follows:

- CAPs 1-5 - in the CEMs on the left and the DEM

- CAP 6 - in the Base (the only required CAP)

- CAPs 7-10 - in the CEMs on the right

If CAPs are not present, ACSLS reports them as "not installed". This keeps CAP locations from changing in SL3000 when new CAPs are added.

CAPs can be dedicated to a partition

■ SL3000 Modules

There are four types of modules in an SL3000 library:

- Base module, one, *required*

  The Base Module consists of a single frame. This module centralizes the infrastructure for all other modules in the library, which includes the power configurations, robotics (TallBot), electronic modules, cartridge access port, storage cells, tape drives, and operator controls.

  One Base Module, and only one, is required for every library installation.

  On the front of this module is:

- A single, 26 cartridge-dual magazine-cartridge access port (CAP).

- A service door for library access.

- A front panel with three LEDs: Library Active, Service Required, and Wait.

- An optional feature for a touch screen operator panel or window.

- Drive expansion module (DEM) one, *on the left side only*

  A Drive Expansion Module (DEM) may be attached adjacent to the Base Module on the left side only. This module allows further expansion of tape drives and provides additional data cartridge capacity.

- Cartridge expansion module (CEM) *left or right side*

  The Cartridge Expansion Module (CEM) provides additional cartridge cell capacity and growth—no tape drives are in this module A maximum of four CEMs are supported on a single library in addition to the Base Module (required) and optional drive expansion module.

- Parking expansion module (PEM) *leftmost* and the *rightmost side* modules.

  The Parking Expansion Module (PEM) is the same as the cartridge expansion module with six columns of arrays (three on the front wall and three on the rear wall) that are inaccessible to allow the library to park a defective robot. In the left PEM, it is the three leftmost columns (front and back). In the right PEM, it is the three rightmost columns.

■ New Panel Types

Instead of having specific panel definitions in the SL3000 that define each possible cell location, there are generic panel definitions.

The three generic panel types map to a variety of specific panel maps. They are:

- A single SL3000 cell panel type.

  To find out which cells are present in this library, as well as which are part of the licensed capacity and are assigned to this partition, you must audit the library.

- A drive panel type.

  ACSLS identifies which drives are present by configuring the library. Before you do this, you must make sure all drives are powered-on and ready before configuring the library.

- Not installed

  This is a placeholder for the front and rear panels on modules that are not yet installed.

Using audit

To find what cells are present in the library you must audit the library. Auditing informs ACSLS which cells are part of the partition that it is managing in this library, as well as, displaying the cells that are actually present on drive panels. To do this:

1. Configure an SL3000 to ACSLS.

2. Audit the library.

3. Look at the allocated and free cells by entering any of the following:

   - `free_cells.sh -a`

   - `get_license_info`

   - `display cell * -f status`

4. Look at the drives using either:

   - `query drive all`

   - `display drive * -f type serial_num wwn`

■ Aliasing SL3000 CAP IDs

The SL3000 has 12 CAPs, but some backup applications currently only support a maximum of 3 CAPs per LSM. CAP aliasing lets you alias CAPs 3-10 as CAPs 0, 1, or 2.

Map the Library CAP IDs to alias CAP IDs by using the following template.

`$ACS_HOME/data/external/SL3000/SL3000_CAP_Aliases.SAMPLE`

Note: When Setting CAP IDs:

For Library CAP IDs:

- LSM must be zero (0)
- CAP ID must be 3-10
- There can be no duplicate library CAP IDs

For Alias CAP IDs

- CAP number must be 0-2
- There can be no duplicate alias CAP IDs

```
# SL3000 CAP Aliases (SL3000_CAP_Aliases)

#

# This file maps real, library SL3000 CAP IDs to CAP ID aliases.
# It is used when ACSAPI clients do not support SL3000
# CAP IDs greater than 2.
# (The SL3000 can have 12 CAPs, with a maximum CAP number of 11.)

#

# Alias a library CAP to an alias CAP number by providing the full #
library CAP ID and the aliased CAP number on the same line.
# For example:
#  Library CAP ID     Alias CAP Number
#  0,0,6              1

#

# Using the CAP Aliases file:
#   (1) A # in the first column comments out a line.
#       The above CAP alias example is commented out.
#   (2) Only specify CAP numbers as an alias when there
#       is no actual CAP at that location.  For example,
#       use CAP 2 as an alias when there is no CAP 2
#       installed in the SL3000 library.
#   (3) CAP numbers 1 and 2 can always be used as aliases.
#   (4) Initially, the CAP number 0 (zero) for the
#       Access Expansion Module (AEM) can be specified as
#       an alias.
#       However, when the SL3000 supports AEMs, you will
#       only be able specify CAP numbers 1 & 2 as aliases.
#   (5) After updating the CAP Aliases file, you must:
#       a) Shutdown ACSLS.
#       b) Reconfigure ACSLS (using acsss_config) to update
#          the CAPs defined in the ACSLS database.
#       c) Restart ACSLS.
# Save the CAP alias file as:
#   $ACS_HOME/data/external/SL3000/SL3000_CAP_Aliases
# by copying and modifying the template:
#   $ACS_HOME/data/external/SL3000/SL3000_CAP_Aliases.SAMPLE

#

#  Library CAP ID     Alias CAP Number
   0,0,4              1
   0,0,6              2
```

This figure shows an example of a modified SL3000_CAP_Aliases file. The comment (#) character has been removed from the lines shown in bold to specify that alias CAP number assignments are enabled. In this example, the:

- physical CAP ID 0,0,4 is aliased to CAP number 1.
- physical CAP ID 0,0,6 is aliased CAP number 2.

■ Moving Cartridges

The SL3000 can partition down to the drive and cell level.

WARNING: If cells are reassigned from one partition to another, the cartridges in those cells will be orphaned, and will no longer be accessible by the partition that they were in before. The host managing the other partition could write over the data on the cartridges.

To prevent cartridges from being orphaned when partition boundaries change: before re-partitioning the library move them to cells that will remain in the partition. Since SL3000 is a single LSM, the existing ACSLS move command does not work, since you would just move them somewhere else in the library. You might also move them to another cell that will also be removed from the partition.

Use one of the following methods to move your cartridges:

- Use the StorageTek Library Console™ (SLConsole).

  You must audit the libarary to audit your volume's locations.

  Refer to the *SL3000 User's Guide* for detailed information and procedures.

- Use the following ACSLS procedure:

  1. Use "volrpt" on page 175 or "display volume" on page 304 to display volume locations.

  2. Display a list of available (empty) cells in a specific panel using the display command:

     ```
     display cell  a,l,p,*,* -status empty -f status
     ```

     For more information, refer to "display cell" on page 293.

  3. Move the cartridges to a specific cell by specifying a free cell instead of an LSM ID. For a cell move, use the move command:

     ```
     move AAAAAA  a,l,p,r,c
     ```

◼ SL3000 CAP Behavior

The SL3000 supports up to 12 Cartridge Access Ports (CAPs). There can be a CAP on each SL3000 panel.

For CAP behavior in a partitioned library, refer to .

WARNING: All ACSs must have at least one CAP. This can be **dedicated** or is shared. Because all of the CAPs in an SL3000 can be dedicated to other partitions, this is a crucial restriction when configuring SL3000 partitions.

◼ Finding MissingSL3000 Cartridges

If a cartridge is out of place or unaccounted for by ACSLS:

1. Perform a physical audit of the SL3000 using the SLConsole.

   The physical audit of the SL3000 is performed as a background task in between handling mount and other library operation requests.

   Warning: If the SL3000 contents are out of sync with ACSLS due to manual operations such as loading cartridges directly, it is not advisable to attempt continued operations.

2. Run an ACSLS audit to update the ACSLS database to match the actual inventory of library cartridges.

◼ Varying the SL3000 Offline

You should vary SL3000 components offline to ACSLS before they are powered off, if they are inoperative, and before you open an SL3000 access door. This notifies ACSLS that they are unavailable. Once they are available, vary them back online.

Use ACSLS to vary SL3000 components offline, not SLConsole

You should vary SL3000 components (ACSs, LSMs, and CAPs) offline to ACSLS, not the SLConsole.

ACSLS allows outstanding requests to complete before taking components offline, unless it is a vary offline force. The SLConsole has no knowledge of outstanding requests to ACSLS.

Varying components offline using SLConsole may cause requests in progress to fail.

When to vary SL3000 components offline to ACSLS

Before opening the access door

Before opening the SL3000 access door vary the ACS offline.

```
vary acs acs_id offline
```

Note: If any CAPs in the SL3000 are in automatic mode, you must:

1. Set them to manual mode before opening the access door.

2. Set them back to automatic mode after you close the access door and the SL3000 comes back online.

If a CAP is inoperative

If a CAP is inoperative, vary it offline:

```
vary cap cap_id offline
```

■ Using the Dynamic Configuration (config) utility

The dynamic configuration (config) utility allows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration changes are recorded in the `acsss_config.log` file.

The following dynamic configuration utilities are supported:

- `config acs`
- `config drives`
- `config lsm`
- `config ports`

Using the `config` utility provides the following benefits:

- ACSLS can continue running, allowing you to perform mount requests to unaffected library components.

- Allows you to reconfigure specified library components while all other configuration information remains unchanged. For example, when specifying:

    - An ACS, the configurations of other ACSs are not affected.

    - An LSM, the configurations of other LSMs are not affected.

    - A drive panel, the drives on a panel, mounts and dismounts to all existing drives are not affected.

■ Auditing New Cartridge Addresses

After an expansion has occurred.

1. Vary the ACS and ports offline.

2. Do one of the following.

   - Add or remove a module(s)

   - Change capacity

   - Re-partition the library

3. Vary the ACS and ports online.

4. Update the configuration in the ACSLS database using one of the following procedures.

   - Dynamic:

     a. `config acs` *acs_id*

        or

        `config lsm` *lsm_id*

     b. Audit the library.

   - Static:

     a. Shutdown ACSLS: `kill.acsss`

     b. Update the ACSLS configuration: `acsss_config`

     c. Bring up ACSLS: `rc.acsss`

     d. Audit the library

■ ACSLS Dual TCP/IP

Dual TCP/IP connections between the ACSLS server and the library is an option that can be purchased for the SL3000.

SL3000 Requirements

- Please coordinate with both your system administrator and network administrator to understand your current network environment and to identify all necessary IP addresses in advance.

- Please coordinate with your system administrator to either configure your network interface or to validate that it is configured properly.

- Verify you have the following:

  - ACSLS 7.3 for Solaris 10 Update 4 for Solaris/SPARC and Solaris/x86

  - ACSLS 7.3 for AIX 5.3 TL6

Refer to "ACSLS Dual TCP/IP Support " on page 344 for more information on this feature and the *StreamLine Modular Library System Dual TCP/IP Feature* document.

Note: the procedure for inplementing Dual TCP/IP connections is the same for both SL8500 and SL3000 libraries.

Library Partitioning

<div style="text-align: right;">**D**</div>

The SL8500 and SL3000 support partitioning of their library.

- The SL8500 Modular Library System provides the ability to partition the library to support from one to four physical partitions.

  For more information on partitioning, refer to the *SL8500 Best Practices Guide*.

  Partitioning the SL8500 library provides the following capabilities:

  - ACSLS can manage open systems drives and cartridges, and HSC can manage mainframe drives and cartridges sharing the library

  - More than one instance of ACSLS can access the library.

    One instance of ACSLS can manage multiple partitions in the same library.

  - Provides protection and isolation of data cartridges in separate partitions.

  - A higher level of data organization.

  - An increase in user efficiency.

- The SL3000 supports up to eight partitions.

  For more information, refer to the *SL3000 User's Guide.*

  The SL3000 can partition down to the drive and cell level.

  WARNING: **You must audit an SL3000 after it is configured or re-configured to ACSLS.** The library reports the potential cell locations that are not accessible to ACSLS when ACSLS audits it. Inaccessible cell locations include:

  - locations where CAPs, drives, and operator panels are installed.

  - cell locations that the robots cannot access.

  - cell locations that are not licensed.

  - cell locations that are not in this partition.

WARNING: If cells are reassigned from one partition to another, the cartridges in those cells will be orphaned, and will no longer be accessible by the partition that they were in before. The host managing the other partition could write over the data on the cartridges.

To prevent cartridges from being orphaned when partition boundaries change, before re-partitioning the library, you need to move them to cells that will remain in the partition. Since SL3000 is a single LSM, the existing ACSLS move command does not work, since you would just move them somewhere else in the library, and you might move them to another cell that will also be removed from the partition.

- To move cartridges to a specific cell you need to specify a cell ID instead of an LSM ID.

    - For a cell move, use:

      ```
      move AAAAAA  a,l,p,r,c
      ```

    - For a standard LSM move, use:

      ```
      move AAAAAA  a,l
      ```

- To display a list of available (empty) cells in a specified panel, use the display command:

    ```
    display cell  a,l,p,*,* -status empty -f status
    ```

■ Guidelines for Partitioning

There are several steps that you need to follow when you are partitioning a library. They are:

1. Plan your new configuration.

2. Schedule an outage to partition or re-partition a library.

   When a library is re-partitioned, the library drops its communication to all hosts managing all partitions, even if their partitions are not affected by the change.

3. Have your service representative license and enable partitioning on the SL8500 or SL3000.

4. Move cartridges from locations where they would not be accessible to locations that will still be in a partition managed by ACSLS.

5. Create your partitions using the SLConsole.

   Refer to the SLConsole help for instructions.

6. Activate partitioning using ACSLS as described in "Partitioning a Library or Changing Partition IDs" on page 370.

Plan Your New Configuration

Plan your new partitioned configuration ahead of time.

- Organize the cartridges and drives for performance.

- If you will do a lot of enters into and ejects from a partition, an SL8500 partition should include at least one of the bottom three rails. Partitions with few enters and ejects can be on the top rail.

Minimize Disruptions when Partitioning an SL8500 or SL3000

When an SL8500 or SL3000 is partitioned or the partitions in an SL8500 or SL3000 are modified, access to your data is affected. Tape drives and cartridges that were managed by one host server, may be managed by a different host.

Changes in Configuration

Changes in configuration occur when a:

- Library's configuration has changed

- Library has been re-partitioned and the partition managed by ACSLS has changed (a message is displayed)

- Library capacity has changed and the cells reported as available to ACSLS have changed

- When LSM Configuration or capacity have changed

When the configuration may have changed, you are notified by ACSLS Event Log and Event Notification messages.

Keeping Outages to a Minimum

The following procedures provided below ensure that outages are minimized and that you can continue to access your data. These procedures explain how to:

- Partition an un-partitioned SL8500 or SL3000 into two or more partitions.

- Modify the partitions in an SL8500 or SL3000 that is already partitioned.

- Merge a partitioned SL8500 or SL3000 so it is a single un-partitioned ACS.

Challenges

The biggest challenges arise when: LSMs (SL8500 rails) are removed from an SL8500 partition; or cells and drives are removed from an SL3000 partition.

- If you are combining multiple SL8500 rails into fewer rails, you need to move the cartridges off the rails (LSMs) that will be removed from the partition to the LSMs that will remain in the partition. This applies whether you are partitioning a library that was not partitioned before or you are re-partitioning a partitioned library.

- Determine how many cartridges you have in the LSMs that you are emptying, and how many free cells are in the LSMs to which you are moving cartridges.

  Use **free_cells.sh -a** to find out numbers of cartridges in these LSMs (allocated cells) and free cells.

- Shortly before you change the library configuration, move the cartridges away from the LSMs that will be removed from the partition.

  Refer to the procedures for "Moving Cartridges Before Removing an LSM from an ACS" on page 161 for details.

■ Partitioning a Library or Changing Partition IDs

These procedures describe how to partition an un-partitioned library or to change partition IDs on an existing partitioned library.

Note:

1. Move the cartridges before removing an SL8500 LSM or SL3000 cells from an ACS.

   If ACSLS manages one of the partitions in the new library configuration, you must first follow the procedures for "Moving Cartridges Before Removing an LSM from an ACS" on page 161 to move the cartridges off of LSMs that will be assigned to another partition.

2. Quiesce the library before re-partitioning it and vary the ACS(s) and port(s) offline.

3. Partition the library using the StorageTek Library Console™ (SLConsole).

4. The following then occurs after you press **<Apply>**. The library:

   - Rejects any new requests from ACSLS.
   - Drops all host connections.
   - Applies partitioning changes.
   - Accepts new host connections.

5. Reconfigure ACSLS to activate the partition changes.

   Because the library was partitioned for the first time or it was re-partitioned and the partition ID changed, ACSLS cannot reconnect to the library. In order to reconnect, you must:

   a. Shutdown ACSLS using `kill.acsss`.

   b. Run `acsss_config` to configure the new partition and select Option 8.

      A message displays asking if the ACS is in a partitioned library. Refer to "Configuring or Updating your Library Hardware" on page 46 for more information.

6. Start ACSLS with the new configuration.

7. Vary the port(s) and ACS(s) online.

■ Re-partitioning a Library

These procedures describe how to re-partition an existing partitioned library.

1. Move the cartridges before removing an LSM or cells from an ACS.

   If ACSLS manages one of the partitions in the new library configuration, you must first follow the procedures for "Moving Cartridges Before Removing an LSM from an ACS" on page 161 to move the cartridges off of LSMs that will be assigned to another partition.

2. Quiesce the library before re-partitioning it and vary the ACS(s) and port(s) offline.

   This causes ACSLS to reject new requests from clients while the ACS(s) remain available for dynamic re-configuration (after the re-partition).

3. Re-partition the library using the SLConsole.

   The following then occurs after you press **<Apply>**. The library:

   • Rejects any new requests from ACSLS.

   • Drops all host connections.

   • Applies partitioning changes.

   • Accepts new host connections.

4. Reconfigure ACSLS to activate the partition changes.

   Note: There may be no changes to a specific partition. The changes may only affect other partitions. In this case, no ACSLS configuration changes are needed.

Note: Because the library was not partitioned the first time or it was re-partitioned and the partition ID did not change, ACSLS can reconnect to the library.

To do this, select one of the following:

- Run Dynamic Config (`config acs` *acs_id*) while ACSLS is running. This updates the ACSLS configuration to match the new library configuration. To do this:

  a. Vary the port(s) online.

  b. Update the ACS(s) configuration with `config acs` *acs_id*.

- Shutdown ACSLS and run `acsss_config` (option 8) to update the partition configuration in ACSLS. To do this:

  a. Shutdown ACSLS.

  b. Update the configuration using `acsss_config.`

  c. Restart ACSLS.

  d. Vary the port(s) online.

5. Vary the ACSs online.

■ Changing a Partitioned ACS to an Un-partitioned ACS

1. Optionally, eject cartridges from the partition(s) managed by ACSLS.

   If ACSLS will not be managing the non-partitioned library, you may want to eject cartridges from the partition(s) managed by ACSLS. This allows you to enter them into other ACS(s) that ACSLS manages.

   You can use the *ejecting.sh* utility for ejecting multiple cartridges.

2. Quiesce the library before re-partitioning it and vary the ACS(s) and port(s) offline.

3. Re-partition the library using the SLConsole.

   The following then occurs after you press **<Apply>**. The library:

   - Rejects any new requests from ACSLS.

   - Drops all host connections.

   - Applies partitioning changes.

   - Accepts new host connections.

4. Reconfigure ACSLS to change the ACS to an un-partitioned library or to remove this ACS (partition) from its configuration.

Because the library was changed from partitioned to un-partitioned, ACSLS cannot reconnect to the library. In order to reconnect, you must:

a. Shutdown ACSLS

b. Run *acsss_config* to configure the new partition.

Refer to for more information.

5. Start ACSLS with the new configuration.

6. Vary the port(s) and ACSs online.

■ Viewing an ACS Partition ID

To view the partition ID(s) for one or more ACSs use one of the following commands:

- `query lmu all` for all ACSs

- `query lmu acs_id` for a single ACS

Output Example:

```
ACSSA> q lmu 0
2008-02-27 06:08:02

ACS:   0      Mode: Single LMU          Master Status:  Not
Comm
Not Partitioned                         Standby Status:  -

      ACS State    Desired State
      offline      online

Port    Port State  Desired State   Role  CL  Port Name
0, 0    offline      online          -    13   172.27.2.6
```

■ CAP Behavior

Partitioned Library

WARNING: All ACSs must have one CAP that is **dedicated** or is shared. Because all of the CAPs in an SL3000 can be dedicated to other partitions, this is a crucial restriction when configuring SL3000 partitions.

Dedicating a CAP in a Partitioned Library

With the SL3000 library, a CAP can be dedicated to a partition. When CAPs are dedicated to a partition, ACSLS can set these CAPs to automatic mode.

Sharing CAPs in a Partitioned Library

In a partitioned SL8500 library, CAPs are always shared among all partitions.

In a partitioned SL3000 library, CAPs can be shared between partitions. The process is as follows:

1. When a host uses a CAP to enter or eject cartridges, it reserves the CAP for its exclusive use.

2. The host must complete the enter or eject and leave the CAP empty and closed.

   Should you need to terminate an enter on:

   - ACSLS - the cancelled `enter` command must be issued

   - HSC - the `drain` command must be issued

3. The CAP is now available for use by any partition in the library.

When Another Host Needs the CAP

If a host does not complete the enter or eject, the CAP may still contain cartridges for the affected partition. The CAP is still in a reserved status. This causes problems when other partitions need to use the CAP. To resolve this, the library must identity the host holding the reserve and terminate the enter or eject from that host.

In rare cases, you may not be able to access the host holding the reservation, either because the host crashed and cannot be restarted or you lack physical access or the security authority to issue the commands needed. In these cases, the CAP reservation can be overridden using the SLConsole.

CAUTION: Overriding a CAP reservation using the SLConsole should be done only as a last resort.

Reserving the CAP

Consider the following when reserving the CAP:

- When a CAP is in use for an enter or eject operation, all 39 slots (SL8500) and 26 slots (SL3000) are reserved for that operation. The CAP cannot be subdivided.

- For addressing purposes, the CAP needs a location (ACS,LSM,CAP#). The LSM number is associated with the second rail in each library. For example: ACS#,1,0 for CAP A and ACS#,1,1 for CAP B

- Reserving the CAP

  The CAP is a common component among partitions. Each host will:

  - reserve the CAP to use it for an enter or eject

- release the CAP when the enter or eject is terminated. Another host can then use it.

Terminating a CAP Reservation

1. When ACSLS finds that a CAP is reserved by another partition, it reports the partition ID and host ID holding the reservation.

   You then need to contact the host owning the reserved CAP and request them to complete the enter or eject.

   Once this is done, the empty CAP is then unreserved and available.

2. If the CAP reservation cannot be released by the host holding it, the SLConsole command is used to override the CAP reservation.

   a. The SLC command displays a warning to the host informing it that the reservation from the specific partition owner will be removed.

   b. The SLConsole queries the CAP to find the partition owner/requester/reserver.

   c. The SLC displays the partition name (HLI1, HLI2, HLI3, HLI4, or Default) in the warning.

3. The library sends a CAP Overridden Message to the host that held the reservation.

   This host should terminate the enter or eject task that was in progress.

4. The library now owns the CAP.

   The CAP must be empty and closed before it is available to be used by another host. If the:

   • CAP has cartridges in it, you must empty the CAP before a partition can use it. Use the SLConsole to do this.

   • CAP is open, you must empty and close it. Then the library can audit it and ensure that it is empty. Until the CAP is closed, it cannot be made available to any partitions.

   • CAP is empty, it will be locked, and it is available to be reserved by any partition.

5. The CAP is now unreserved and is available for use by any partition.

316120101

ACSLS Support of the SL500

<div style="float:right">**E**</div>

The SL500 library is a SCSI-attached, single-LSM library. It requires either a fibre channel or a LVD SCSI connection to the ACSLS server for control path operatios. This chapter discusses ACSLS support for the SL500 library.

■ ACSLS Connectivity

The ACSLS server needs a Host-Bus Adapter (HBA) that is compatible with the interface card in the SL500 library (either LVD SCSI or Fibre). Refer to the ACSLS Product Information Guide for available feature codes and part numbers for library connections.

When you run `$ACS_HOME/install/install_scsi_sol.sh`, the system is probed for the existence of new hardware: either a JNI card or an LSILogic card. If either of these HBA cards is present on the system, but the corresponding driver is not in place, the routine displays a prompt that asks you for permission to locate the appropriate driver package and install it.

■ ACSLS and SL500 Library Differences

Library Configuration

- ACS, LSM - SL500 library is an ACS with a single LSM (1 ACS and 1 LSM)

- SL500 does not currently support the pass-thru-port

- SL500 has at least 1 base module, and can have up to 4 expansion modules.

- ACSLS considers each SL500 module as a panel.

- SL500 libraries managed by ACSLS reports 6-character volser labels through ACSL interface

Identifying Library Locations

Addressing schemes

ACSLS addressing is ACS, LSM, panel, row, column

- SL500 addressing is LSM within the library, module, row, column

- ACSLS addresses are zero-based. They start with panel 0, row 0, and column 0.

- SL500 libraries are one-based. They start with module 1, row 1, and column 1.

- In order to ensure consistency between the SL500 library and ACSLS internal addresses, ACSLS defines placeholders for panel 0, row 0, and column 0.

Panels

- Panel 0 = not installed

- Panel 1 = base module

- Panels 2-5 = expansion modules

Row numbers

- Within each module (panel), rows are numbered 1-12

- Rows are numbered consecutively from the top down

- Numbering is identical between the SL500 and ACSLS

Column numbers

- Numbered from 1 to 11 (maximum)

- The current SL500 configuration is limited to 9 columns

- Columns 1-4 are on the left side from front to back

- Columns 5-8 on are the right side from back to front

- Column 9 is the back wall where the drives are located

- Numbering is identical between the SL500 and ACSLS

Drive addresses

SL500 addressing is - LSM, module, row, column

- Row is equal to the drive number or slot number

- Drive number is 1-2 on the base module and 1-4 on the expansion modules

- Column number is always 9 for drives

- ACSLS addressing is ACS, LSM, panel, drive

- Drive number is equal to a row or slot

Figure 29. Example of drive addressing

	ACSLS Drive Identifier	SL500 Drive Address
Base module	0, 0, 1, 1	0, 1, 1, 9
	0, 0, 1, 2	0, 1, 2, 9
Drive expansion module	0, 0, 2, 1	0, 2, 1, 9
	0, 0, 2, 2	0, 2, 2, 9
	0, 0, 2, 3	0, 2, 3, 9
	0, 0, 2, 4	0, 2, 4, 9
Next expansion module	0, 0, 3, 1	0, 3, 1, 9
	etc.	etc.

■ ACSLS Limitations

- Maximum number of SL500 libraries = 31

- Maximum number of LSMs = 127

- Maximum number of panels in a SL500 library = 5

  - Expansion modules add 1 panel each

- An SL500 with the base module only has just 1 panel

  - The "move" command cannot be used

- The SL500 firmware does not support unlabeled cartridges. This means that the "venter" command cannot be used

■ Configuring SL500 Library Settings

The SL500 supports new configuration settings that affect ACSLS. They are:

- CAP as storage – CAP cells

  - Only allowed in expansion modules

  - Base module CAP is always used as a CAP

  - Configured on a per-module basis

  - Any CAP used as storage cannot be used in enter/eject operations

- Reserved cells are for library use only (diagnostic, cleaning cartridges)

  - Only affects base module

- Number (n) can be configured from 0 to 9 (max row in base module)

- The first (n) cells in column 1 will be reserved

- These cells are considered inaccessible to ACSLS

■ SL500 CAP Behavior

The CAP in the base module has one five-slot magazine. When a drive expansion module is added, the CAP in the drive expansion module has two five-slot magazines. In the case of expansion modules, the CAP can be configured for 10 additional storage slots per expansion module. In addition:

- CAP cells, other than those on a base module, can either be configured as CAP cells or as storage cells.

- The base module CAP is always used as a CAP.

- The SL500 must have one active CAP for entering and ejecting cartridges.

- If there are multiple CAPs, the SL500 library always reports to ACSLS that it has one CAP.

- The doors to all of the CAP sections lock and unlock as one unless the CAP has been configured as a storage cell.

  If a module has been configured to use CAP cells as storage cells, the CAP section is not affected by library operations that access the CAP.

- When ACSLS audits the CAP, it examines all cells.

■ Auditing the Library

You must always perform an audit if:

- This is a new library

- One or more modules have been added, removed, or swapped.

- Cartridges have been manually added or removed through the door

- After library configuration settings have changed

Note: A self-audit of the library occurs when you:

- Power-on or re-initialize the library

- Open and close the door

After the self-audit is complete you need to update your database using ACSLS.

When this is a new library

A new library must be audited to synchronize the actual contents of the library with the database.

After adding, removing, or swapping modules

After a SL500 module has been added, removed, swapped, or if you are auditing the library for the first time, perform the following steps:

1. Vary the SL500 library offline to ACSLS and power down the library.

2. Perform the library change, such as adding a module.

3. Power on or re-initialize the SL500 library.

4. Terminate ACSLS (if it is running).

   ```
   idle
   kill.acsss
   ```

5. Run `acsss_config.`

6. Start up ACSLS.

   ```
   rc.acsss
   ```

7. Perform an audit of the SL500 panel(s) affected.

After manually adding or removing cartridges through the door

If you have manually added or removed cartridges from the LSM through the library door you must synchronize the databases. To do this, perform an audit as described under "audit" on page 188.

Note: After entering or ejecting cartridges through the CAP

If you have entered or ejected cartridges from the LSM through the CAP using the ACSLS `cmd_proc`, the database is automatically updated.

After changing library configuration settings

If you have changed library configuration settings you should reconfigure ACSLS using acsss_config and then audit the library. You should also reconfigure and audit the library after changing the following library settings:

- Number of reserved cells

- CAPS as storage cells

To do this:

1. Vary the SL500 library offline to ACSLS and power down the library.

2. Perform the library change.

3. Re-initialize the SL500 library.

4. Terminate ACSLS (if it is running).

   ```
   idle
   kill.acsss
   ```

5. Run `acsss_config.`

6. Start up ACSLS.

   rc.acsss

7. Perform an audit of the SL500 panel(s) affected.

Troubleshooting

<div style="float:right">

F

</div>

This chapter summarizes tools, tips, and techniques for troubleshooting problems in ACSLS. The range of troubleshooting resources includes logs, key observation points, and diagnostic probes.

■ ACSLS Event Log

The ACSLS event log is the first stop for useful information in the event of problems with your library operation. This log contains information about library events, status changes, and errors. All sub-components within ACSLS will report events to the acsss_event.log by sending messages to a process called the event logger. The standard event log, which is automatically created when ACSLS is installed, is contained in the file `$ACS_HOME/log/acsss_event.log and` where `$ACS_HOME` is usually `/export/home/ACSSS/`.

Logged events include:

*   Significant Events

    These are normal events that can help you manage the library. For example, events are logged when an audit is initiated or terminated, a device changes state, or a CAP is opened or closed.

*   Library Errors

    These are events where both fatal and nonfatal hardware and software errors are logged. Examples include: LSM failures; problems with cartridges; database errors; process failures; and library communications failures.

Each message in the event log includes a time stamp, the name of the component reporting the message, and a description of the event. For a complete explanation of each message, consult the *ACSLS Messages* manual.

A window on the ACSLS console displays a running tail of the event log. You can generate a similar display from any shell window. To do this:

1. As user `acsss`, run the command

    `acs_tail $ACS_HOME/log/acsss_event.log`

2. To view the entire event log, use a text editor, such as vi, that enables you to navigate through the log, search for specific errors, or follow specific sequences of events.

Managing the Event log

ACSLS continues sending messages to the acsss_event.log.

- When this file reaches a threshold size (500KB by default), the file is renamed to `event0.log` and saved in the log directory. The `acsss_event.log` then continues as a new file.

- When `acsss_event.log` again reaches the threshold size, the `event0.log` is renamed to `event1.log` and the `acsss_event.log` is renamed `event0.log`.

- This process continues for as many log files that are configured for retention.

By default, nine event log files are retained in the log directory. With each subsequent threshold, the oldest file is removed and all remaining files are sequentially renamed.

You can configure the maximum size of the acsss_event.log and the number of log files to retain using `acsss_config`, Option 2. Refer to "Setting Event Logging Variables" on page 36.

Using *greplog* to Search Event Logs

The diagnostic tool, *greplog,* allows you to perform keyword searches through any and all event log files. Used very much like the Unix *grep* utility, *greplog* will return the complete log message associated with a given keyword expression. This allows you see the message's date and time stamp, message number and the function text related to every message containing that expression.

Format

```
greplog [-iv] pattern file_1 file_2 ... file_n
```

Options

- -i instructs *greplog* to ignore the case of the search pattern expression.

- -v instructs *greplog* to filter out all messages containing the expression and to display all of the entries in the log file. The exception are those entries which match the pattern expression.

pattern: the pattern is the search criteria to be used.

```
file_1 file_2 ... file_n
```

greplog accepts multiple file parameters and wild card expressions in the file list.

Examples

- To display all occurrences within an event sequence, use the sequence number.

  ```
  greplog 1392 acsss_event.log
  ```

- To search the Event Log for all messages about volume CART89:

  ```
  greplog CART89 acsss_event.log
  ```

- To search all archived copies of the Event Log for messages about tape mounts:

  ```
  greplog -i mount event*.log
  ```

Additional Logs

The `acsss_event.log` contains all of the messages pertaining to any aspect of the ACSLS running processes. But there are additional files in the log directory that contain status information about external utilities, such as backup and restore and installation utilities.

- S87ACSLS.log - Contains messages pertaining to the startup of ACSLS at boot time.

- acsss.pid - Stores the process id of the currently running acsss_daemon.

- acsss_config.log - Contains a summary of each library configuration.

- acsss_config_event.log - Contains event messages that were posted by the acsss_config routine.

- bdb_event.log - Contains event messages that were posted by the database backup utility, bdb.acsss.

- cron_event.log - Contains messages that were posted by cron utilities. (To see the cron schedule, run the command `crontab -l`.)

- di_trace.log - Contains trace information related to the database interface.

- install.log - Contains event messages posted while running the installation script, install.sh.

- ipc_trace.log - Contains trace information pertaining to ACSLS inter-process communications.

- rdb_event.log - Contains event messages that were posted by the database restore utility, rdb.acsss.

- timed_bkup.sh.log - Contains event messages related to the automatic database backup utility.

Additional trace logs may be found in the log directory depending upon the specific tracing that you have enabled on your system. These include the following:

- acsss_stats.log - Volume statistics tracing is enabled by acsss_config.

- acsss_trace.log - Client-server tracing is enabled at the request of Sun Software Support personnel.

- acslh.log - Host-LMU tracing is enabled at the request of Sun Software Support personnel.

- scsilh.log, mchangerX.log, scsipkt.log - All of these contain traces of SCSI communications to a SCSI-attached library and they are enabled at the request of Sun Software Support.

Trace Log Management

Trace logs that are enabled at the request of Sun Software Support can grow quite rapidly. These logs need to be monitored and managed in order to mitigate problems of a full disk.

The utility *monitor.sh* is provided to perform automatic log management and archiving services. The syntax is

```
monitor.sh <name of log>
```

When this utility is enabled to monitor a specific log, it will allow the log to grow to a size of 1MB (default) and then will compress the log using *gzip,* placing the compressed logfile with a time stamped name in the `ACSSS/log/log_archives` subdirectory. This operation will continue as long as tracing stays enabled.

■ Key Observation Points

There are numerous utilities that enable you to verify the status of various aspects of ACSLS.

- psacs - This utility shows a summary of all of the ACSLS running processes. It is the best indication whether ACSLS is running or not. A typical output should display no fewer than twelve different processes, all children of a common parent process.

- db_command status - This is a quick check to see whether the PostgreSQL database is running

- get_license_info - This is a quick check to confirm that the product is not inhibited due to a license issue.

- pkginfo -l STKacsls - This is a quick check to verify the version of ACSLS you are running.

■ Testing Library Connections

The first step to isolate a communication problem with an attached library is to test the connection with a simple packet connection. To test any library connection, you must first bring down ACSLS in order to free the port for access by the diagnostic utility. There are numerous commands available to use in the appropriate context.

The command *testports* test each configured connection between ACSLS and the attached libraries. The syntax is:

```
testports
```

For each defined port in ACSLS, *testports* displays the port name and attempts to communicate to the library behind that port. It then displays the result. This test should reveal any library that is not responding.

Different library types are connected to ACSLS by means of different communication media. ACSLS provides tools to test a host-library connection for each library type. Each of these commands requires that you specify a port address. If you do not know the port expression to use, you can discover it using the following query:

```
sql.sh -d "select port_name from portable"
```

Once you know the port address, you can test the following:

- For TCP/IP libraries, including the SL8500 and 9310*:*

  ```
  testlmutcp <ip address:port>
  ```

- For serial libraries, including the 9310, 9315, 4410 and 9840*:*

  ```
  testlmu <tty port name>
  ```

- For SCSI libraries, including fibre:

  ```
  probescsi.sh <mchanger port name>
  ```

- For fibre-attached libraries:

  ```
  probeFibre.sh
  ```

  No port name is required since this utility will discover all fibre-attached libraries:

To test the connection to every configured mchanger device:

```
showDevs.sh
```

This utility reveals the HBA in the connection and displays the target, LUN, and model number of each attached library that responds. In addition, you can:

- To reveal the World Wide Name of the mchanger device:

  ```
  showDevs.sh -w
  ```

- To reveal the serial number of the mchanger device:

  ```
  showDevs.sh -s
  ```

■ Testing a Client Connection

Client applications communicate with ACSLS over TCP/IP using the RPC (remote procedure call) protocol. If a client system is unable to communicate with ACSLS, you can use *rpcinfo* to test whether ACSLS is reachable from the client machine. To do this:

1. From the ACSLS server, verify that ACSLS is running.

   ```
   psacs
   ```

2. From the ACSLS server, verify that the RPC daemon is running.

   ```
   ps -ef | grep rpc
   ```

3. From the ACSLS server, verify that program number 300031 is registered for TCP and IDP.

   ```
   rpcinfo | grep 300031
   ```

   This program number confirms that ACSLS is running and that ACSLS has registered with RPC.

4. From the client machine, or any Unix machine on the network, use *rpcinfo* to exchange a packet with program number 300031 on the ACSLS server.

   You must specify the IP address of the ACSLS server along with the program number.

   ```
   rpcinfo -t <ip address> 300031
   ```

   If the communication exchange was successful, the *rpcinfo* utility will display the message

   ```
   program 300031 version 1 ready and waiting
   program 300031 version 2 ready and waiting
   ```

   This confirms that ACSLS is available for client connections across the network.

■ Gathering Diagnostic Information for Sun Software Support

As part of the service call, your software support representative may ask you to send the entire set of diagnostic logs and other diagnostic information for analysis by Sun. All of this data can be collected with a single command:

```
get_diags
```

When this utility has collected all of the information, it prompts you to either email the data or to make it available for manual transfer.

If you elect to email the data directly from the ACSLS machine, make sure that email communication is possible between your ACSLS machine and the Internet. Your enterprise may have a firewall to prevent email going directly from the target machine. In this case, you can email the information to yourself within the enterprise and then forward the diagnostic data to Sun.

Alternatively, you can elect to transfer the information manually. The *get_diags* utility advises you where to find the waiting tar packages for transfer. Typically, the staging area for diagnostic data is `/export/backup/diag/acsss`.

Improving Library Performance

<div style="float:right">G</div>

■ Overview

This chapter discusses how to maximize your library performance during pass-thru activity between Library Storage Modules (LSMs). ACSLS works to minimize pass-thru activity in several ways. You can enhance performance by using these ACSLS facilities to minimize pass-thru activity. They are:

- Minimizing pass-thru activity between LSMs

- Floating cartridges during a dismount

- Entering cartridges into a library

- Ejecting cartridges from a library

- Maintaining empty storage cells in a library

- Minimizing pass-thru activity with ACSAPI requests and ACSLS commands

- Using these ACSAPI requests and ACSLS commands

■ Minimizing Pass-thru Activity Between LSMs

To improve tape library performance, it is important to minimize pass-thru activity from one library to another. When cartridges and drives are in connected libraries, it is always desirable if tape mounts occur when both the drive and the cartridge are in the same LSM (library). Mounts that require fewer pass-thus result in better performance.

■ Floating Cartridges During a Dismount

When cartridges are dismounted, ACSLS tries to avoid pass-thru activity among LSMs by assigning a new home cell whenever the cartridge's old home cell is in a different LSM than the transport.   This is ACSLS's default behavior. To disable this feature, use the "Extended Store" feature for an LSM. Cartridges being dismounted are "floated" to a new home cell to avoid movement between LSMs.

■ Entering Cartridges into a Library

When cartridges are entered into a library, each cartridge is assigned a home cell in the closest LSM to the CAP. If the LSM with the CAP has empty cells, home cells are assigned in this LSM. If the LSM with the CAP is full, home cells are assigned in the closest LSM with empty cells.

To optimize performance, enter cartridges into an LSM with tape drives that are compatible with the cartridge(s). For example, enter 9940 cartridges into an LSM with T9940B tape drives, not into an LSM with only T9840B tape drives. Otherwise, the 9940 cartridges must be passed-thru to an LSM with T9940B tape drives.

■ Ejecting Cartridges from a Library

When cartridges are ejected from a library, the cartridges must be moved to the specified CAP(s).

To optimize performance, eject cartridges to a CAP in the LSM where the cartridges are located. If the cartridges are located in several LSMs, choose a CAP that is close to most of the cartridges.

■ Maintaining Empty Storage Cells in an LSM

ACSLS cannot "float" cartridges to an LSM on a dismount if all the storage cells are full. Similarly, when cartridges are entered into a full LSM, the cartridges must be passed-thru to an LSM with empty cells.

To optimize performance, you should identify cartridges that do not need to be kept in the library for StreamLine access and eject those cartridges from the library. You may also want to move cartridges from full LSMs to other LSMs with enough empty cells so that all LSMs have empty storage cells.

Notes: Managing the cartridges in a library should be done when the library is not busy with mount requests.

■ Minimizing Pass-Thru Activity Using ACSAPI Requests and ACSLS Commands

These ACSAPI requests and ACSLS cmd_proc commands help you to avoid unnecessary pass-thru activity between LSMs.

The following applies to both ACSAPI requests from a client to ACSLS, and to ACSLS commands entered using cmd_proc. The term "request" is used for both ACSAPI requests and cmd_proc commands.

mount scratch and mount *

For a specified transport, select a scratch cartridge and mount it. Optionally, select the cartridge from a specified scratch pool and/or with a specified media type. ACSLS selects a compatible scratch cartridge in the closest LSM to the transport. In order to rotate use among all cartridges, the compatible cartridge with the least recent access date is selected within the selected LSM.

Command:

ACSAPI	`mount scratch`
cmd_proc	`mount * drive_id [pool_id]` `[media media_type]`

- For an ACSAPI request, `drive_id`, `pool_id` and `media_type` must be specified. (Either a specific media type, `ALL_MEDIA_TYPE`, or `ANY_MEDIA_TYPE` can be specified for `media_type`.)

- With a cmd_proc `mount *` command, if `pool_id` is not specified, it defaults to the common pool (pool 0).

  A specific media type can be identified. If media * is specified, `ANY_MEDIA_TYPE` is chosen. If media is not specified, `ALL_MEDIA_TYPE` is chosen.

The special `media_type` values of `ANY_MEDIA_TYPE` and `ALL_MEDIA_TYPE` are processed as follows:

- When `ALL_MEDIA_TYPE` is specified, a cartridge with a media type compatible with the transport is selected. (This is based on the media_compatibility file.)

- When `ANY_MEDIA_TYPE` is specified, the scratch_preferences file identifies the preferred list of media to mount on a transport.

query mount

For a specified cartridge (`vol_id`), return a list of transports that are compatible with the cartridge's media type, ordered by proximity to the cartridge. (The compatible transports in the closest LSM are listed first, etc.)

Command: `query mount vol_id`

query mount scratch and query mount *

For a specified scratch pool, a list of all transports compatible with the media type(s) of the cartridges in a specified scratch pool is returned. A specific media type can be specified to restrict the transports to only those compatible with the specified media type.

The list of transports returned is ordered so the transports closest to the densest scratch pools are listed first.

Command:

ACSAPI	`query mount scratch`
cmd_proc	`query mount * ` *`pool_id`* `[media ` *`media_type`* `]`

Either a specific media type or `ALL_MEDIA_TYPE` (media * in a cmd_proc command) can be specified. `ANY_MEDIA_TYPE` is **not** supported.

Note: When `ALL_MEDIA_TYPE` is specified, cartridges with media compatible with the transport are selected to determine the densest scratch pool.

■ Using these ACSAPI Requests & ACSLS Commands

This section describes how to use the ACSAPI requests and commands for improving tape library performance.

In the following discussion the term "request" is used for both ACSAPI requests and cmd_proc commands.

When mounting a specific cartridge (where the vol_id is known):

- Precede the `mount` request with a `query mount` request.

- Pick the first "available" transport, and specify this transport in the `mount` request.

When mounting a scratch cartridge on a specific transport, there are two options:

- To select the closest scratch cartridge to a specific drive:

  Use a `mount scratch` request specifying a transport and, optionally, a scratch pool.

  For ACSAPI requests, specify one of the following:

  - `ALL_MEDIA_TYPE` (compatible media is selected)

  - `ANY_MEDIA_TYPE` (uses the scratch preference list).

  For cmd_proc commands, select one of the following:

  - Do not specify a media type (compatible media is selected)

  - Specify `media *` (uses the scratch preference list).

- To select a specific drive from a list of drives and then select the closest scratch cartridge:

  Pick the transport, then pick the scratch tape: This ensures the transport is close to the scratch cartridge.

  a. Enter a `query mount scratch` request to identify the closest available transport to the most scratch media in the desired scratch pool.

  b. Enter a `mount scratch` request to mount a scratch cartridge on the selected transport. Optionally, specify the scratch pool.

  For ACSAPI requests, either specify:

  - `ALL_MEDIA_TYPE` (and compatible media will be selected), or

  - `ANY_MEDIA_TYPE` (to use the scratch preference list).

  For cmd_proc commands, either:

  - Do not specify a media type (and compatible media will be selected), or

  - Specify `media *` (to use the scratch preference list).

Firewall Security Option

H

■ Overview

The firewall-secure option allows you to run ACSLS behind a firewall while the client software makes requests across that firewall.

Firewall security is also offered to ACSLS clients, which also allows them to operate behind their own respective firewalls. This is made available by Sun StorageTek to its ISVs (Independent Software Vendor partners). Please contact the ISV for your client software component to find out the latest status for each specific client.

■ Running ACSLS behind a Firewall

This Firewall-secure solution provides the following capabilities:

- Allow ACSLS to run behind a firewall (i.e. ACSLS on the secure side of firewall, client on opposite side)

- Allow ACSLS client(s) to run behind their own firewall(s) (i.e. client(s) on secure side, ACSLS on the opposite side of firewall).

  Important: To do this, the ISV must have implemented the supplied changes within their client side software component.

- Preserve compatibility with current ACSLS client implementations, allowing those clients to continue to run with ACSLS in the firewall solution.

- Preserve current ACSAPI/Client functionality and performance. This includes all functionality that is available in a non-firewall environment.

A complete solution would include combining the first two capabilities above. This would allow ACSLS and the ACSLS client(s) to each run behind their own respective firewalls (i.e. two firewalls between ACSLS and the client(s)), and still have the same communications performance as within a non-firewall environment.

■ Addressing security areas

ACSLS has addressed the following security concerns as follows:

RPC

The use of RPC within ACSLS is a concern for some sites in trying to run within a firewall environment. Preserving compatibility with the current installed client base precludes the ability to remove RPC completely from the ACSLS.

However, the ACSLS firewall-secure feature has addressed the concerns inherent in RPC, which are:

- The need to allow outside (untrusted) parties to initiate connections to the trusted host across an unrestricted range of ports (1024-65535).

- The exposure of the mapping of available services on a platform via the portmap (or rpcbind) daemon running on well-known port 111.

Security

In a firewall solution, the fundamental security comes from restricting access from the untrusted (non-secure) side into the trusted (secure) side. In all cases, some limited and controlled access must be allowed in order to perform communications and allow data exchange. The goal is to allow that data exchange within a well-defined and restricted set of entry points, allowing you to control those access points and their corresponding communications. This goal is met by this solution.

Communications Components

ACSLS/Client communications rely on two network interface components to handle network communications between client platforms and the ACSLS platform. Software which acts as a client or proxy-server for ACSLS implements one of these two components, in order to be compatible with ACSLS platforms and existing clients. The component residing on the client platform is known as the SSI; the component residing on the ACSLS platform is known as the CSI. While it would be desirable to implement all changes within one side (e.g. the ACSLS platform), in order to maintain client compatibility and to provide all the firewall-secure features it is necessary that corresponding changes be made to each side where the benefits are to be realized. The positive side of this is that each side can independently implement the features and realize the firewall-secure benefits on its side (e.g. changes to the ACSLS allow the ACSLS platform to run behind a secured firewall).

■ Benefits of the Firewall-Secure Option

Server Side

With changes to just the server side component, as provided within this Firewall-secure solution, the benefits are:

- Restricts incoming connections for ACSLS communications to a single TCP port for all registered program numbers (there are two registered program numbers for the ACSLS CSI, both of which will be serviced by one single port).

- Allows users to specify the identity of that port, and configure their firewall in a corresponding fashion.

- Allows users to turn off ACSLS communications to UDP ports.

- Allows users to disable any communication by the ACSLS server to the **client-side** portmapper(s)* (UDP/TCP port 111). The portmapper must still remain running on client platforms to preserve compatibility with client side code. However, it will not be used for network communications initiated by the server, and therefore the clients' firewall(s) can be configured to disallow access to it.

- Outgoing connections from the ACSLS server side to the client(s) are unrestricted with respect to the server-side ports used to preserve current performance. This follows the widely accepted practice by the security community.

Server Port Restriction

This solution potentially restricts the number of incoming ports to which any outside party can communicate with the server platform (i.e. initiate network connection) to either one or three ports - the single customer-specified port for ACSLS incoming requests, plus possibly the two portmapper ports (TCP & UDP port 111).

Note: To disallow client access to the **ACSLS server** portmapper, and thus disallow access to UDP and TCP ports 111, the changes must be made to the **client** software component. See the client side discussion below.

The server side of the solution, above, is implemented completely within this firewall-secure ACSLS solution.

Client Side

An identical set of restrictions can be realized on the client side. This would give the client an identical capability of residing behind its own secure firewall, as described in the server side discussion above. These restrictions allows the user to:

- Restrict incoming connections for communications (response) to the client to a single TCP port for each registered program number. There is one registered program number for the ACSLS SSI.

- Specify the identity of this port, and configure the firewall in a corresponding fashion.

- Turn off client-side communications to UDP ports.

- Disable any communication by the client to the **ACSLS server** portmapper (UDP/TCP port 111). The portmapper must still remain running on the ACSLS platform to preserve compatibility with ACSLS code. However, it will not be used for network communications initiated by the clients, and the ACSLS server firewall can be configured to disallow access to it.

- Outgoing connections from the client side to the server are unrestricted with respect to the client-side ports used to preserve current performance. This follows widely accepted practices by the security community.

Client Port Restriction

This solution restricts the number of incoming ports to which any outside party can communicate with the client platform (i.e. initiate network connection) to between one and three ports - one specified port for client incoming responses, and possibly the two portmapper ports (TCP & UDP port 111).

Note: To disallow ACSLS server access to the **client's** portmapper (and thus disallow access to UDP and TCP ports 111), the changes must be made to the **ACSLS server** software component (see server side discussion above).

This solution has a two-step implementation:

- Sun StorageTek has made the needed code changes to the ACSLS CDK source code (CSC Developer's Toolkit)

- Clients of ACSLS who wish to provide this security for their client platform must integrate these changes into their client-side SSI code, rebuild that product, and again certify their CSC (Client System Component) with Sun StorageTek's ACSLS.

Advantages

Each of the client-side and server-side parts of the solution is independent. This means that if only one of the two sides is behind a firewall with respect to the other side, the solution (software changes) need only be implemented on that side. In addition, that partial solution maintains compatibility with all current client and server implementations which already exist, as well as compatibility with other software components which use the CSI / SSI interface. This includes compatibility with current Sun StorageTek products. This solution does not impact current performance, with respect to client / server communications.

■ Firewall-Secure Solution Scenarios

The following diagrams show possible scenarios of the operation, port usage, and relationship of the ACSLS components when used across a firewall. They are intended to be understood in conjunction with the text just presented (above). The "SSI" in the following diagrams is the network interface component of ACSLS that runs on the client side of the communications. The CSI is the network interface component of ACSLS that runs on the ACSLS platform.

Example 1

In Figure 30, firewall security is implemented on the ACSLS server side (CSI) only.

Figure 30. Firewall security on ACSLS server side (CSI) only

CLIENT   FIREWALL(S)   SERVER

SSI ————————————————————————————————▶ CSI

Source Ports: Unrestricted Ports

Destination Ports: 3 Static Ports
1 - Designated
(Fixed single port set on CSI)
2 - TCP 111
3 - UDP 111

SSI ◀———————————————————————————————— CSI

Destination Ports: 3 Static Ports
1 - Dynamic Single Port*
2 - TCP 111**
3 - UDP 111**

Source Ports: Unrestricted Ports

S403_006
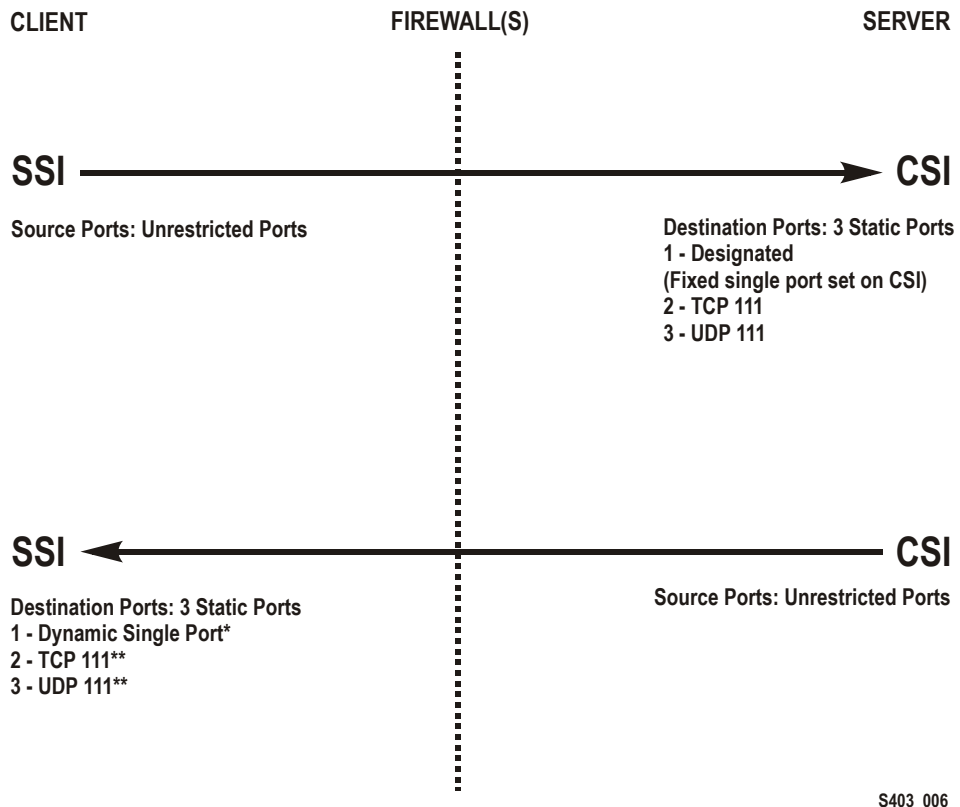
Notes:

* Dynamic in this example means that the port is selected by the SSI at startup from the range 1024-65535. The port is not designated by the user, nor is it typically the same port across new executions of the SSI (i.e. from one instance of an SSI running process to the next).

** The portmapper 111 port(s) on the SSI side is **only rarely** queried by the CSI. It is only accessed by the CSI in the case where the return port number provided by the SSI in its request packet does not function (i.e. results in a network interface failure) for sending the response packets back to the SSI. In this case, as a retry mechanism, the CSI queries the SSI-side portmapper for the port to use, which is registered with the portmapper under the SIE's program number.

Example 2

In Figure 31, firewall security is implemented on the client side (SSI) only.

Figure 31. Firewall Security on Client Server Side (SSI) Only

```
CLIENT                          FIREWALL(S)                          SERVER

SSI ──────────────────────────────┊──────────────────────────▶ CSI

Source Ports: Unrestricted Ports                    Destination Ports:
                                                    1 - Dynamic Single Port*
                                                    2 - TCP 111
                                                    3 - UDP 111


SSI ◀──────────────────────────────┊────────────────────────── CSI

Destination Ports: 3 Static Ports                   Source Ports: Unrestricted Ports
1 - Designated
(Fixed single port set on SSI)
2 - TCP 111**
3 - UDP 111**
```

S403_007
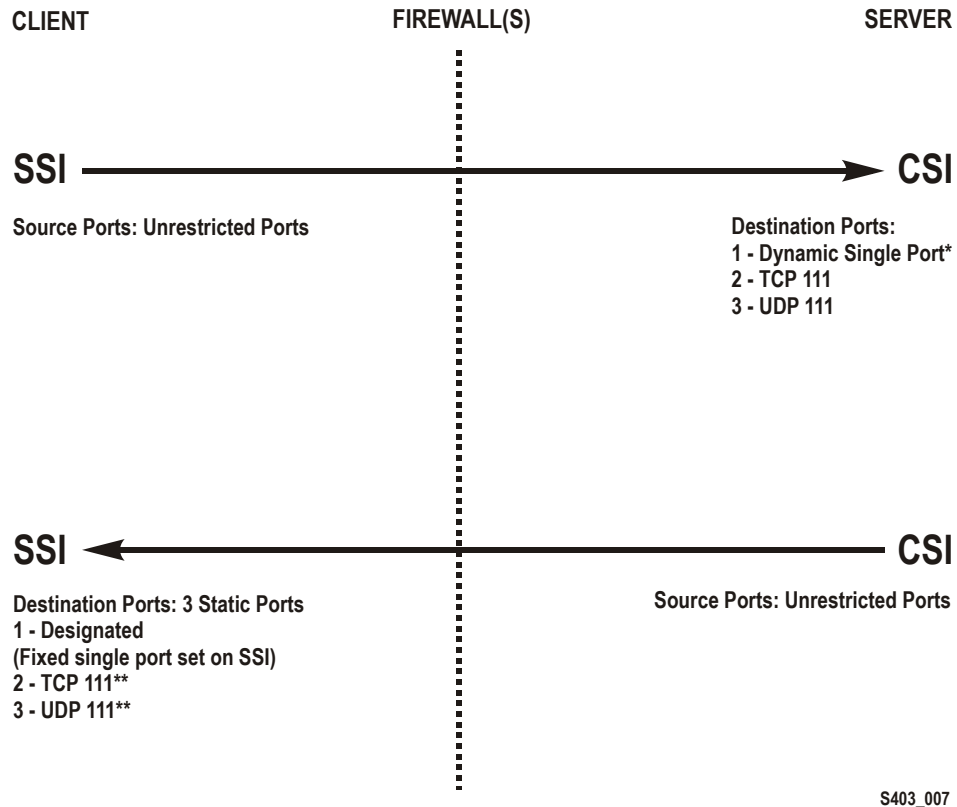
Notes:

* Dynamic in this example means that the port is selected by the CSI at startup from the range 1024-65535 and the port is not designated by the user, nor is it typically the same port across new executions of the CSI (i.e. from one instance of an CSI running process to the next).

** The portmapper 111 port(s) on the SSI side is only rarely queried by the CSI. It is only accessed by the CSI in the case where the return port number provided by the SSI in its request packet does not function (i.e. results in a network interface failure) for sending the response packets back to the SSI. In this case, as a retry mechanism, the CSI queries the SSI-side portmapper for the port to use, which is registered with the portmapper under the SSI's program number.

Example 3

In Figure 32, both client (SSI with CDK 2.2 and Firewall-secure Supplement) and the ACSLS server (CSI) are implementing Firewall-Secure API's. The client and server is still relying on the portmapper for port identification).

Figure 32. Firewall Security on Client (SSI) and ACSLS Server (CSI) using Portmapper

CLIENT · · · FIREWALL(S) · · · SERVER

SSI ——————————————→ **CSI**

Source Ports: Unrestricted Ports

Destination Ports: 3 Static Ports
1 - Designated
(Fixed single port set on CSI)
2 - TCP 111
3 - UDP 111

SSI ←—————————————— **CSI**

Destination Ports: 3 Static Ports
1 - Designated
(Fixed single port set on SSI)
2 - TCP 111*
3 - UDP 111*

Source Ports: Unrestricted Ports

S403_008
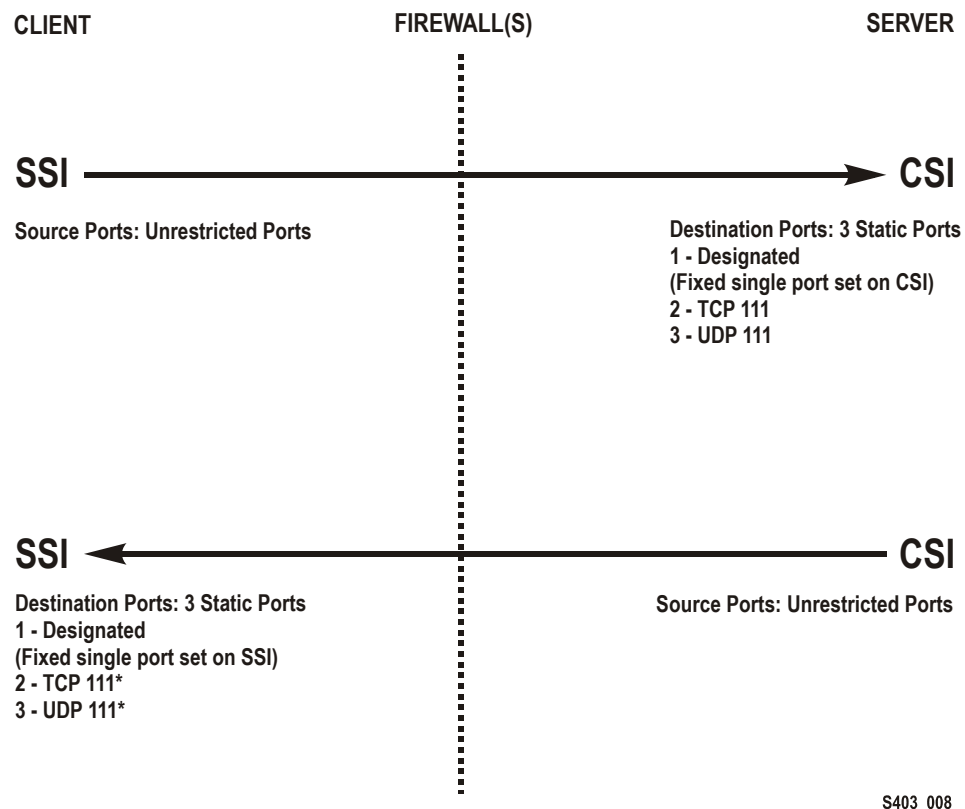
Notes:

* The portmapper 111 port(s) on the SSI side is only rarely queried by the CSI. It is only accessed by the CSI in the case where the return port number provided by the SSI in its request packet does not function (i.e. results in a network interface failure) for sending the response packets back to the SSI. In this case, as a retry mechanism, the CSI queries the SSI-side portmapper for the port to use, which is registered with the portmapper under the SSI's program number.

Example 4

In Figure 33, both Client (SSI with CDK 2.2 and Firewall-secure Supplement) and ACSLS Server (CSI) implementing Firewall-Secure API's (Client and Server have implemented the "No Portmapper" feature).

Figure 33. Firewall Security on Client (SSI) and ACSLS Server (CSI) Without the Portmapper Feature

CLIENT   **FIREWALL(S)**   **SERVER**

SSI ──────────────────────────────────────► **CSI**

Source Ports: Unrestricted Ports

**Destination Ports: 1 Static Port
1 - Designated
(Fixed single port set on CSI)**

SSI ◄────────────────────────────────────── **CSI**

**Destination Ports: 1 Static Port
1 - Designated
(Fixed single port set on SSI)**

Source Ports: Unrestricted Ports

S403_009
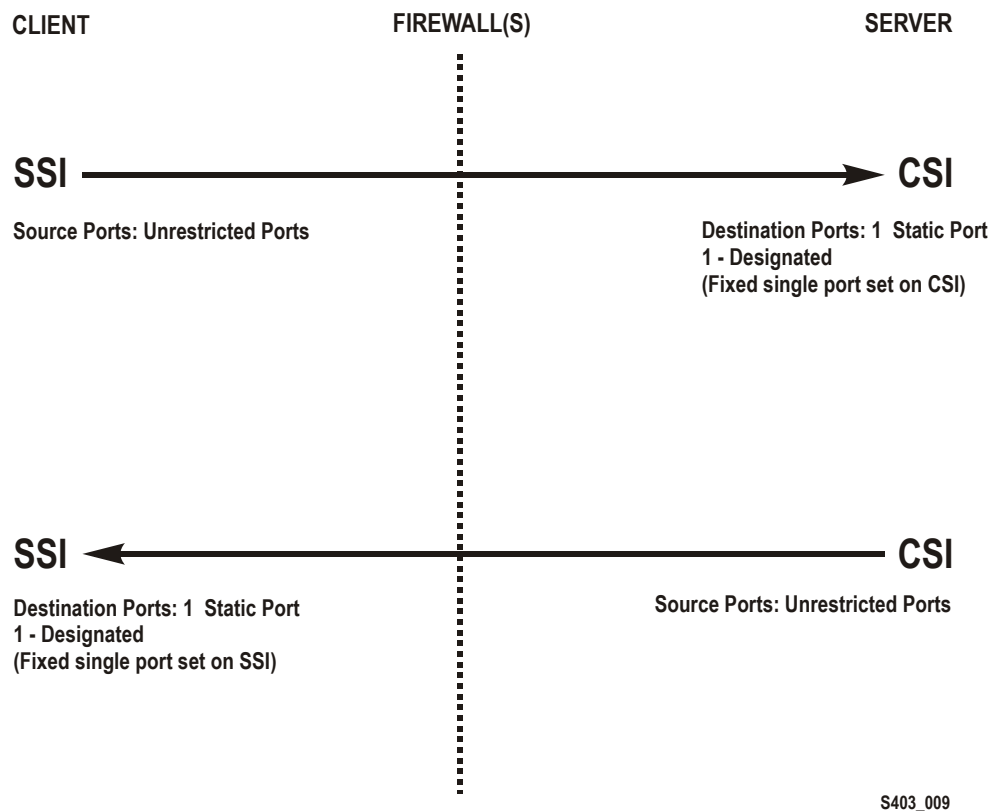
■ Turning On the Firewall-Secure Option

To turn on the firewall-secure option, you need to set several variables using the `access_config` utility.

1. Log in as `acsss`.

2. At the password prompt, enter the password for user `acsss`.

3. Stop the ACSLS server

    Note: You must shut down the ACSLS server for the new firewall-secure variables to take effect.

    `kill.acsss`

4. To run the configuration script, enter

    `acsss_config`

    The ACSLS feature configuration screen appears.

5. Select option 1 - `Set CSI tuning variables`

    Accept the default for all variables except for the following.

    a. Set the value to `TRUE` at the following prompt:

        `Changes to alter use of the TCP protocol will not take effect until the product is restarted. CSI support for RPC using the TCP protocol is enabled [TRUE].`

        Variable: `CSI_TCP_RPCSERVICE`

        Turning on TCP insures that the TCP protocol is available for use by clients of ACSLS for network communications. The firewall-secure feature of ACSLS supports TCP only, so clients should perform network communications using this protocol.

    b. Set the value to `FALSE` at the following prompt:

        `Changes to alter the use of the UDP protocol will not take effect until the product is restarted. CSI support for RPC using the UDP protocol is enabled [TRUE].`

        Variable: CSI_UDP_RPCSERVICE

        CAUTION: **You must ensure that no ACSLS clients are depending on this UDP protocol. The firewall-secure ACSLS runs on TCP only.**

        Turning off UDP insures that no clients will access the server using this protocol. This allows you to disallow all general UDP access to the ACSLS platform at the firewall, allowing only those accesses which are specifically required in your environment.

        You **must** allow clients access to the UDP and TCP port 111 for portmapper access, unless those clients implement the firewall-secure feature, and specifically turn off their queries to the ACSLS portmapper.

    c. Set the value to `NEVER` at the following prompt:

        `Changes to alter use of the port mapper will not take effect until the product is restarted. Enable port mapper: (ALWAYS / NEVER / IF_DUAL_LAN_NOT_ENABLED) [IF_DUAL_LAN_NOT_ENABLED].`

Variable: CSI_USE_PORTMAPPER

NEVER allows clients of ACSLS to disallow external access to the portmapper on those client platforms.

IMPORTANT: This does **not** allow you to turn off external access to the portmapper on the ACSLS platform; to do that, the client(s) of ACSLS must have adopted the firewall-secure changes in the client software component(s), and this feature must be turned on in the client software component.

This feature ensures that the ACSLS server will not make any queries of the portmapper on the client platform. This allows any firewall **which is protecting the client** to disallow access to the portmapper.

d. Set the value to `TRUE` at the following prompt:

```
Enable CSI to be used behind a firewall (user-
defined inbound port) (TRUE/FALSE) [FALSE]:
```

Variable: CSI_FIREWALL_SECURE

TRUE allows you to specify the single port that ACSLS will use for accepting inbound client communications (i.e. TCP connections). This variable simply enables this feature. The specific port will be specified in the next variable.

e. Set the value to an available fixed port on the ACSLS server at the following prompt:

```
Port number used by the CSI to receive incoming
ACSLS requests.
```

Variable: CSI_INET_PORT

This is the port which will be used by the ACSLS CSI component for accepting incoming network connections.

IMPORTANT: You must configure your firewall to allow incoming connections on this port. This ensures that only that port is exposed for use by those outside clients wanting to initiate communications with ACSLS. You may disallow connections on all other incoming ports except this one, and UDP/TCP port 111 (unless clients have implemented the feature to eliminate their queries to the ACSLS portmapper; in that case, port 111 may also be disallowed at the firewall). The recommended default value for this port is 30031. It is unlikely (but not impossible) that this port will be used by other processes on most systems. See the Appendix F, "Troubleshooting" for steps to take if there is a port conflict.

6. Select `E` to exit `acsss_config`.

7. Restart ACSLS by entering:

```
rc.acsss
```

Your changes are saved.

■ Turning Off the Firewall-Secure Option

Some of the variables used above for turning on the firewall-secure feature are also related to turning off that feature. To turn off the firewall-secure behavior, it is only necessary to perform the steps below, but a specific site may wish to make modifications to other variables as well.

1. Log in as `acsss`.

2. At the password prompt, press enter the password for user `acsss`.

3. Stop the ACSLS server

   Note: You must shut down the ACSLS server for the new firewall-secure variables to take effect.

   ```
   kill.acsss
   ```

4. To run the configuration script, enter

   ```
   acsss_config
   ```

5. Select option 1 - `Set CSI tuning variables`

   Change the following values that were set when you configured the firewall-secure feature. You must change the following variables:

   a. Set the value to `ALWAYS` at the following prompt:

      ```
      Changes to alter use of the port mapper will not
      take effect until the product is restarted. Enable
      port mapper: (ALWAYS / NEVER /
      IF_DUAL_LAN_NOT_ENABLED) [IF_DUAL_LAN_NOT_ENABLED].
      ```

      Variable: CSI_USE_PORTMAPPER

   b. Set the value to `FALSE` at the following prompt:

      ```
      Enable CSI to be used behind a firewall (user-
      defined inbound port) (TRUE/FALSE) [FALSE]:
      ```

      Variable: CSI_FIREWALL_SECURE

6. Select `E` to exit `acsss_config`.

7. Restart ACSLS by entering:

   ```
   rc.acsss
   ```

   Your changes are saved.

■ Firewall-Secure Configuration

The following requires that you are knowledgeable with regard to configuring the network firewall behind which ACSLS resides. ALL firewalls are "third-party" software, and will have varying details with regard to setting them up correctly for protecting your network environment. The following is not meant to be a recommendation of firewall security policy, but rather a set of helpful instructions for what the firewall must / can do with regard to the ACSLS product only. See your System Administrator for other security details.

Here is a list of details for setting up your firewall with respect to the ACSLS platform:

- Put in place an overall rule to disallow UDP incoming and outgoing connections.

- Put in place an overall rule to disallow TCP incoming connections (TCP outgoing connections must remain open).

- Put in place a specific rule to allow incoming TCP connections on the port which you specified for the ACSLS usage. **IMPORTANT:** This port **must** match the one you configured under acsss_config, or you will receive no client communications at the ACSLS server.

If all of your clients have implemented the firewall-secure feature and make no queries to the ACSLS platform's portmapper, you are done. If the clients still make use of that portmapper on the ACSLS platform, you must add the following:

- Put in place a specific rule to allow incoming and outgoing connections on the well-known portmapper TCP and UDP port 111.

Example

The following is an example of the rules which were put in place for an iptables-based firewall in order to put all of the above rules in place.

Note: These are in addition to other rules configured for the specific firewall.

```
echo " - FWD: Allow all connections OUT and only existing/
related IN"
$IPTABLES -A FORWARD -i $EXTIF -o $INTIF -m state --state \
      ESTABLISHED,RELATED -j ACCEPT

# These rules allow client access to the portmapper
$IPTABLES -A FORWARD -p tcp -i $EXTIF --dport 111 -j ACCEPT
$IPTABLES -A FORWARD -p udp -i $EXTIF --dport 111    -j ACCEPT

# These rules allow client access to the ACSLS CSI for network
communication

# Note: This assumes that the CSI firewall-secure port was
specified as 30031
```

```
$IPTABLES -A FORWARD -p tcp -i $EXTIF --dport 30031 -j ACCEPT
# Catch all rule, all other forwarding is denied and logged.
$IPTABLES -A FORWARD -j drop-and-log-it
```

■ Troubleshooting Firewall-Secure Communications

Troubleshooting a network communications interface which includes the ACSLS platform and clients, and now includes intervening firewall(s), may involve multiple steps. By introducing the firewall(s) into the path between ACSLS and its clients, there are more potential causes for network communications failures. Additionally, there are more components that must be configured in a way that corresponds with the settings in other components, and if these settings do not match, the network communications will be impacted. Here is a list of things to check and try if you've done all the configuration work on ACSLS, its client(s), and the firewall(s), and network communications are not working.

1. Checking the ACSLS platform:

   - Is the ACSLS up and running? If not, check the acsss_event.log for possible reasons, or for pointers to a possible culprit.

   - Is the CSI being brought up successfully by ACSLS? If not, there should be informative messages in the acsss_event.log which point towards the cause. Bad values for some of the configuration parameters or a port conflict are likely possible causes.

   - Is there a port conflict being reported in the acsss_event.log which causes the CSI to fail? If so, you should use the "netstat" or similar system utility to tell you which ports are in use on the system, and configure the ACSLS to use an available port. Remember to reconfigure the firewall to match.

   - Is the CSI registering for the port you expect? Use the command 'rpcinfo -p' to look at the portmap table. The CSI is registered under program number 300031. Check to make sure that the port registered under that program number is the one you expect (the default port is 30031, with one less zero than the program number).

2. If ACSLS and the CSI are up and running and correctly registered, the next step would be to check access to the ACSLS platform across the firewall:

   - Is the ACSLS reachable via basic RPC? Use the "rpcinfo -t <hostname> <program-number> <version-number>" command to send a simple RPC request to the CSI. (Use "man rpcinfo" on your system to get more information on the rpcinfo command and its use.) Do this from a machine on the inside of the firewall with ACSLS (e.g.

from the ACSLS platform itself), and from outside the firewall. If it works from inside but not from outside, then the firewall is intercepting your ACSLS requests. You need to double check the configuration of the firewall and the ACSLS port. Also check to be sure that the portmapper is accessible through the firewall (this test cannot be used from outside the firewall if access to the portmapper is disallowed).

- Do the ports configured for ACSLS and for the firewall match? Double-check these parameters. This is a likely cause of failure in network communications. Aside from the configured values, perform the 'rpcinfo -p' command mentioned above to insure that the CSI is indeed registering with the expected portnumber. If it is not, look in the acsss_event.log for information about the cause.

- Is the ACSLS receiving requests, but unable to send back responses? If you check the acsss_event.log and find that the CSI reports many dropped network packets or failures to communicate with network clients, then the client requests are getting in, but the responses are not getting out. Again, this is an indication that they are being blocked by a firewall.

3. If your problems are still not resolved.

The above addresses several levels of things to look for. If these yield no specific answer, it's time to do some lower-level checking to find out where communications are being broken down. The best way to do this is through the use of a network packet sniffer facility, such as 'snoop' under Solaris. Use "man snoop" on your Solaris-based system to get more information on the snoop command and its use.

Similar packet tracing facilities are available on other network-connected systems.

- To utilize this, you will have to do your packet sniffing from locations that show you where the packets are getting to and where they are being lost. This may be from both inside and outside the firewall.

- Additionally, looking at the packet data will be informative. If either side is allowing use of the portmapper, it is likely you will see some PORTMAP packets.

- Also, you should see RPC packets passing between the ACSLS and its clients.

- Finally, looking at the transport-level TCP connection will inform you of the specific ports being used on each side for the connection. This is often critical information to find out where the communications are being stopped.

- More detail on performing these operations is beyond the scope of this manual, but your System Administrator should be able to provide some help in this area.

■ Frequently Asked Questions

Why do I need the firewall-secure solution for ACSLS?

The firewall-secure solution allows you to effectively run the ACSLS behind a firewall, and allows you to restrict ports on that firewall so that security is significantly enhanced.

What releases of ACSLS will support the firewall-secure feature?

Only ACSLS 7.0.0 and above support this feature.

What is the maximum number of ports I will have to leave open if I use this firewall-secure feature?

The maximum number of ports on which you might have to allow incoming network connections is three: one for the ACSLS network interface, and two for the portmapper (UDP and TCP 111). Outgoing ports are unrestricted, in accord with accepted industry security practices.

What is the minimum number of ports I can leave open?

The minimum number is one. This is possible if your clients (ISV software) have also implemented the firewall-secure features in their client, and make no queries to the portmapper which resides on the ACSLS platform. When that is the case, the only port that need be open for incoming connections is the one user-specified TCP port used by the ACSLS network interface.

Why doesn't the feature use a range of ports?

There is no architectural advantage to using a range of ports, and there are some security disadvantages. The non-firewall-secure ACSLS uses a range of ports which consists of the full range of dynamic ports available on any given platform. This is correctly perceived as a potential compromise to the security of a site. Restricting this as much as possible, without adversely affecting ACSLS performance, is the goal in order to eliminate that compromise. Since the ACSLS network interface uses only one incoming port at any given time, there is no reason to extend the range beyond one port, provided that port is dedicated to ACSLS use for the ACSLS platform.

What if the port I choose conflicts with another usage of that port on my system?

This is one of the reasons that the port is made user-specifiable. The specific ports available will vary from one customer site to another. The user is not allowed to use one of the well-known reserved ports from 0-1023. The default port of 30031 falls within the range of registered ports, which makes it less likely (though not impossible) that another application which uses dynamic ports will utilize it. Although it is in the range of

registered ports, there is no application registered to use it, which makes it a reasonable default selection.

Does this feature allow me to protect my ACSLS server with a firewall?

Yes, with this feature in place, your ACSLS server can be put on the trusted side of a firewall, with clients accessing it from the opposite (untrusted) side or from the same side.

Does this feature allow me to protect my ACSLS clients (ISV components) with a firewall?

Potentially, yes, but not by itself. In order to realize this scenario, your client software components (clients of the ACSLS) must have adopted the firewall-secure feature, which has been made available via the Sun StorageTek CDK (CSC Developer's Toolkit). Contact your client software provider for a current update on their status.

If I want to be able to protect my clients with a firewall, what should I do?

You should contact your client software provider. They can tell you whether they have adopted any firewall-secure changes in their CSC (client software component).

What about the portmapper? Can I completely disallow access to the portmapper?

If your clients have adopted the firewall-secure changes, they may allow you to shut off the client's queries to the ACSLS platform's portmapper. In that case, you may disallow access to the portmapper on the firewall which protects the ACSLS platform. In any other case, the clients will depend on the ACSLS server side portmapper to help them make a connection with the ACSLS network interface, and it must be available for their use.

Why must the client implement some changes in order for my ACSLS server firewall to shut down access to the ACSLS platform portmapper?

Because it is the client that is making these queries of the ACSLS platform. If the client continues to make these queries, the ACSLS platform must continue to provide the portmapper's services in order for those queries to succeed.

I think the portmapper is bad. Why didn't you remove it completely?

The portmapper provides an important service to legacy clients. Removing it completely would invalidate the interface on which those clients depend. In short, no legacy clients would work without recoding, retesting, and again certifying with the new non-portmapper interface. In this firewall-secure solution, we have provided the capability to remove the queries to the portmapper from both the ACSLS to the client, and from the client to the ACSLS, but we cannot force client software to conform to this. Thus, the portmapper must remain available at least as an optional service until

a site's clients have adopted the firewall-secure features and no longer make use of the portmapper service.

Some of my clients have adopted the firewall-secure features and some have not. How can I take advantage of this?

Those clients which have adopted these features may be protected behind their own respective firewalls. In addition, access to the portmapper's well-known ports may be restricted at the firewall, and then configured to allow access to the portmapper only by those clients who require it. The details and ability to do this will vary based on the specific firewall in use at the site.

I think RPC is bad. Why didn't you remove it completely?

The ACSLS network interface has been RPC-based since the first release of ACSLS. It has proven to be an effective, stable, and reliable mechanism, offering various advantages at the network communications layer. However, it can also be more difficult to secure a platform which uses RPC, due to its common dynamic allocation of ports and use of the portmapper. In this firewall-secure solution, both of these areas are addressed, which allows the customer to effectively configure a firewall in a restricted fashion, yielding the security benefits for which they have the firewall in place.

Additionally, complete removal of RPC from the ACSLS network interface would invalidate all current (legacy) ACSLS clients, making it impossible for any of them to communicate with ACSLS without recoding, retesting, and again certifying their CSCs (client software components).

How will the firewall-secure feature affect network communications performance and timing between my ACSLS clients and the ACSLS server?

There is no effect on performance due to the new firewall-secure features. The usage of a firewall may have performance implications, but this will be based on the operational characteristics of each specific customer's firewall implementation. With a firewall which has negligible impact on performance, the ACSLS and its clients will continue to perform as they did prior to installing the firewall-secure feature. Also, the ACSLS network interface tolerances can be configured, so that delays imposed by the firewall could be handled gracefully.

How does the firewall-secure feature affect the rest of my ACSLS operations?

There is no effect or impact on other parts of the ACSLS operations due to the installation of the firewall-secure solution.

How does the firewall-secure feature affect the ACSLS functionality that my clients use (via the ACSAPI)?

The full set of functionality that is provided through the ACSAPI (and which our ACSLS clients use today to interface with ACSLS) will operate the same under the firewall-secure feature as it does without the feature installed. In particular, this firewall-secure feature supports access control, and also all of the newer features that have been added to the ACSLS product. The full functionality of the ACSAPI will continue to be supported by this feature.

Does the firewall-secure feature work with the ACSLS HA (High Availability) solution?

The firewall-secure feature does not adversely affect HA operation. However, the HA solution is not designed to be run across a firewall (i.e. with each HA server on opposites sides of a firewall). The HA solution requires remote access to the portmapper, so the firewall could not disallow that access if an attempt were made to run each server on opposing sides of a firewall. There are other details of running across a firewall that could adversely affect an HA setup; it is highly recommended that this not be done.

If the HA servers are set up on the same secured side of the firewall, that set of HA servers could be set up with the firewall-secure feature, and clients on the opposite side of the firewall would be able to interact across the firewall with those servers with the same performance and behavior as they would against a non-firewall-secure HA solution.

Does this firewall-secure feature work with other Sun StorageTek software products?

Interoperability with other Sun StorageTek products, as well as partner products (such as client software components which communicate with ACSLS) has been completely preserved. Those products can continue to operate without modification, communicating with the ACSLS server, with the ACSLS server running behind a secured firewall, or in the same environment with those products (as it does today).

Do other Sun StorageTek software products have the same firewall-secure features?

Other Sun StorageTek products do not gain the firewall-secure benefit simply by being used in the same environment with the firewall-secure ACSLS. Each product can work with a firewall secured ACSLS (see previous question), but putting each of those products behind its own respective firewall is a question that the specific product itself must address. Some Sun StorageTek products already have built-in policies which allow some restriction at a firewall used to protect the platforms where those products run. Additionally, any product which acts as a client to ACSLS has the option of adopting the firewall-secure changes which were made to ACSLS, and which are provided as part of the Sun StorageTek CDK (CSC Developer's Toolkit).

Media Management

<div style="text-align: right">**I**</div>

This chapter discusses how to manage the same type of media recorded in different densities.

■ Overview

Problems

Often, new tape drives are introduced that use existing tape media, but record at a higher density. The new drives can often read tapes recorded in the old density, but they cannot write at the old density. The older tape drives usually can neither read nor write in the higher density.

Because of these restrictions, you can encounter these problems:

*   When a tape written in the new format is mounted on an older drive, the old drive cannot read the tape.

*   Storage management applications often fill up partially used tapes by appending additional files at a later time. This will fail if the tape was written in a different format than the drive uses to record data. When a storage management application tries to append additional files to a tape that was originally initialized in an incompatible format, the mount may fail. In addition, if you load the wrong cartridge in a drive, there is the possibility of overlaying data already on the cartridge.

If you have a mix of both older and newer tape drives in your libraries, you will need to manage your tape cartridges that are the same media type. This document describes how ACSLS pools can be used to separate cartridges written in the old format from cartridges written in the new format.

Examples

The following examples show tape drives that record on the same media in different formats:

*   **9940 Media that is used by both T9940A and T9940B Tape Drives**

    The T9940B tape drive uses the same media as the T9940A, but writes data at double the T9940A's density. The T9940B can read T9940A media and can reformat it for writing double density data, but it cannot append data to a previously written T9940A cartridge.

The T9940A drive can reformat a T9940B cartridge for writing single density data, but can neither read from nor append data to a T9940B cartridge.

- **9840 Media that is used by both T9840A & T9840B and T9840C Tape Drives**

  The T9840A and T9840B tape drives use the same media and record in the same format. The T9840C also uses the same media, but records at double the density. Customers must manage 9840 media when both T9840A/T9840B and T9840C drives are present.

- **SDLT Media that is used by both SDLT 220 and SDLT 320 Tape Drives**

  The SDLT 220 and SDLT 320 tape drives both use the same media, but record in different densities. You must manage the common SDLT media when both types of SDLT drives are present.

Solution

ACSLS has tools that you can use to manage common media that two or more types of drives write on but are in incompatible recording techniques. The client application must use these facilities to manage the read/append data incompatibilities.

You can manage drives that record on common media in different formats within an ACS by either:

- Replacing all of the older drives in an ACS with the new drives at the same time.

  This is the simplest and safest strategy.

  Do not append files to any tapes written in the old format after replacing the old drives with the new drives. (With Veritas NetBackup, this is done by suspending the tapes.)

- Gradually replacing the older drives with the new drives.

  This requires you to manage the common media recorded in different formats by creating separate pools for each format. These can either be managed with ACSLS pools or by using the facilities of a backup application (for example, Veritas NetBackup, Legato NetWorker, IBM Tivoli, or CA BrightStor).

■ Managing Common Media Recorded in Different Densities with ACSLS Pools

The example below describes how to manage 9940 media recorded in both T9940A and T9940B formats. This procedure applies to all cases where different tape drives record on a common media in different formats. Use the following table to apply the procedure to your specific case:

Old Format Drive(s)	New Format Drive(s)
T9940A	T9940B
T9840A or T9840B	T9840C
SDLT 220	SDLT 320

Note: The discussion below uses the following syntax conventions for commands and utilities:

- Commands and Utilities entered as is are in **bold**.

- Variables (you must enter the exact value) are in **bold italics**.

Perform the following:

1. Before installing T9940B drives in an ACS that has T9940A drives, identify all 9940 data (non-scratch) cartridges in the ACS and assign them to a T9940A pool. Assign 9940 scratch cartridges to either the T9940A or T9940B pool.

   a. Define T9940A and T9940B media pools:

   `define pool pool_id`

   b. Report all 9940 media.

   You can do this by using either the `display volume` command (option 1) or a custom `volrpt` (option 2) as shown below:

   Results are written to a file.

   - Option 1

   Use the `display volume` command to write the results to a file.

   `display volume * -media STK2P > filename`

   where:

   All 9940 tape cartridges (media STK2P) are reported.

   `filename` is the filename to which the output is written. The **ACS ID** in which the cartridge is located is also listed.

   Read the output, selecting cartridges from specific ACS(s), if desired.

If you want cartridges from one ACS, use the -home operand to only select cartridges from this ACS.

```
display volume * -home acs_id.*.*.*.*  -media
STK2P> filename
```

- Option 2

  Use the custom `volrpt` for all volumes in the ACS. The results are written to a file:

  ```
  volrpt -d -f custom_volrpt_file -a acs_id > filename
  ```

  where:

  custom_volrpt_file is the filename specifying the fields reported in a custom `volrpt`. The following fields are reported:

VOLUME_ID	6	2
MEDIA_TYPE	7	2
VOLUME_TYPE	4	2

  acs_id is the ID of the ACS that you are managing.

  filename is the filename to which the output is written.

  Read the output, selecting only the volumes with a media type of STK2P.

c. Assign the selected volumes to the appropriate pools.

  Assign all non-scratch (**VOLUME_TYPE = "D"**) cartridges to the T9940A pool.

  ```
  set scratch off pool_id vol_id
  ```

  Assign the scratch cartridges (**VOLUME_TYPE = "S"**) to either the T9940A or T9940B pools.

  ```
  set scratch pool_id vol_id
  ```

2. After the T9940B drives are installed, use the pools to mount cartridges on compatible tape drives. Again, cartridges written at T9940A density can be read by either T9940A drives or T9940B drives, but only T9940A drives can append data on 9940A media. Only T9940B drives can read or append to cartridges written at 9940B density.

   The following commands can be used to identify, select and mount cartridges from the pool appropriate for a T9940A or T9940B drive. Use the pools to ensure you are mounting cartridges on the correct drive.

   If you want to mount a scratch cartridge on the correct drive type, follow procedure A. If you want to mount a non-scratch cartridge with data already recorded on it, follow procedure B.

Procedure A - Mounting Scratch Cartridges

a. Query drive(s) to identify drive types.

`query drive` *drive_id* `| all`

b. Identify the correct pool (media type) for the cartridges to be mounted on the desired drive

c. Mount scratch cartridges to a drive from the specified pool.

`mount *` *drive_id pool_id*

Procedure B - Mounting Non-scratch Cartridges

a. Display the status of drives compatible with a cartridge.

`display volume` *vol_id* `-f pool`

The pool for this volume is displayed.

b. Use the following query command to identify a compatible drive.

`query mount` *vol_id*

Drives compatible with a cartridge are displayed. You must select the correct drive type before mounting the cartridge. This query command returns both T9940A and T9940B drives (because both drive types are compatible with 9940 media). Use drive type to separate T9940As from T9940Bs.

c. Mount the cartridge on the selected drive.

`mount vol_id` *drive_id*

3. When all the data on the cartridge has expired, you can migrate the scratch cartridges to the pool for the new recording format. Because T9940A and T9940B drives can re-initialize cartridges in their density, scratch cartridges can be reassigned to a different pool.

`set scratch` *pool_id vol_id*

4. After all T9940A drives are converted to or replaced by T9940B drives, all 9940 media can be used by T9940B drives as long as data is not appended to cartridges written in T9940A format.

CSCI

<div style="text-align: right; font-size: 2em;">J</div>

This chapter describes the Client Server Communications Interface (CSCI) component.

This chapter covers the following topics:

- Overview of CSCI
- Error messages
- Environmental variables

Installation of CSCI should follow installation of the operating system and installation and configuration of ACSLS.

Note: For ACSLS to function properly, the START_CSCI_Process configuration variable must be set to TRUE. You do this in Option 1 of acsss_config. See "Setting Variables that Control ACSLS Behavior" on page 31.

■ Overview of CSCI

The Client Server Communications Interface (CSCI) is a stand-alone component responsible for the passing of request and response packets between a client system and a server system. CSCI is an optional component that lets MVS or RMLS/CSC clients communicate with ACSLS. If you do not use CSCI, you use CSI by default to communicate with ACSLS.

CSCI Connectivity

CSCI provides IPC and TCP/IP connectivity to link the ACSLS server with MVS clients or RMLS/CSC clients

Architecture and Subsystems

Figure 34 on page 424 shows an overall CSCI architecture and subsystems.

Figure 34. CSCI Architecture and Subsystems

S403_010
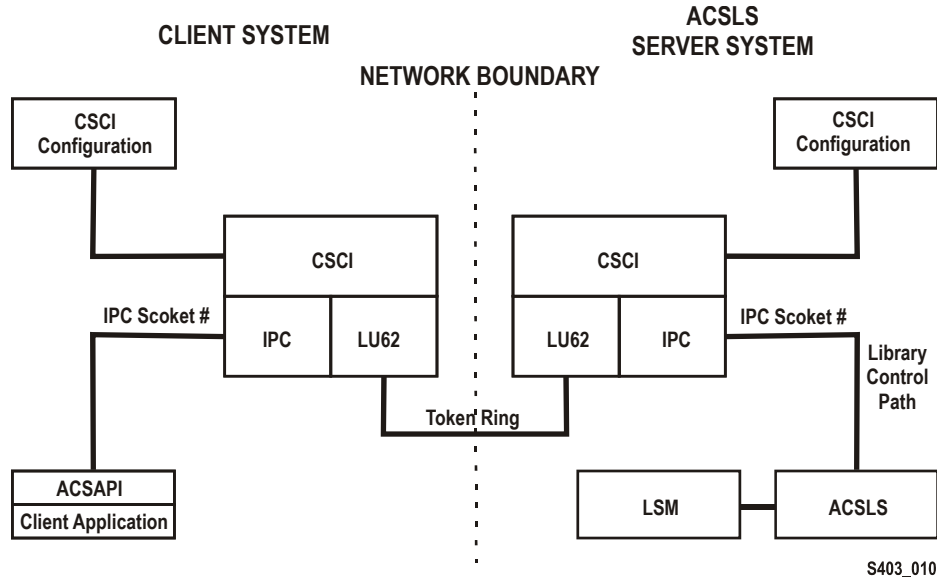
Figure 35 shows how the CSCI system connects with the network interface. On the server side, the data flows from the network hub through the TCP/IP interface to the CSCI interface. From here the data flows through an IPC interface to the ACSLS software in both directions but in a half duplex mode, i.e., one direction at a time.

Figure 35. CSCI System with Token Ring Interface

S403_011
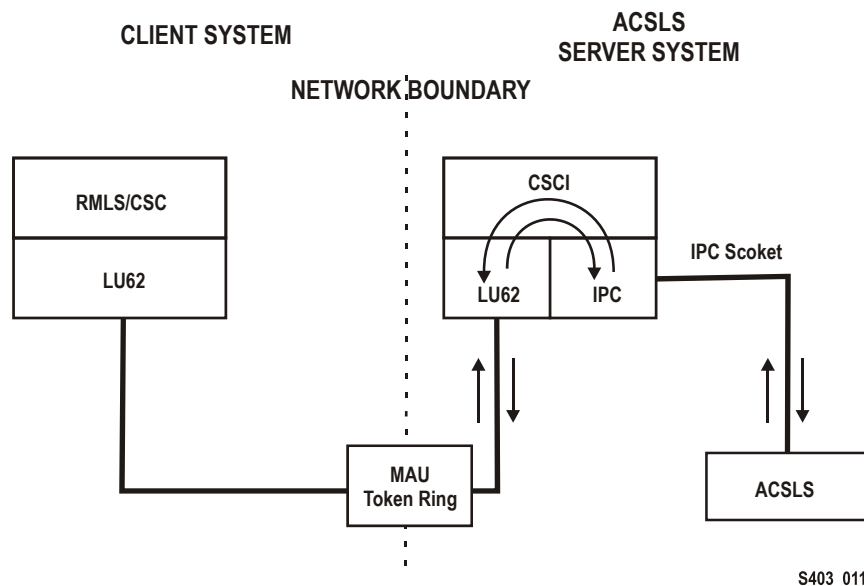
■ Error Messages

CSCI installation and configuration error messages relate to the following conditions:

- Improper ownership

- Improper access permission

- Undefined environment variables

- A variety of miscellaneous conditions

- Improper CSCI and TCP/IP configuration

Common Error Messages

`Communication failure, verb = `*`cmroutine`*`, rc = `*`xx`*`, errno = `*`yyy`*` IPC failure on Socket CSCI`*`zzzzz`*

Explanation: The read failed on the named connection.

Variable:

- *xx* is IBM CPI return code

- *yyy* is the UNIX errno value in /usr/include/sys/errno.h

- `CSCI`*`zzzzz`* is the socket number that was in use

- *cmroutine* is the IBM CPI routine name that failed

`Read failed for connection: `*`connection_name`*

Explanation: The read failed on the named connection.

Variable: connection_name is the name of the connection on which the read failed.

`Write failed for connection: `*`connection_name`*

Explanation: The write failed on the named connection.

Variable: connection_name is the name of the connection on which the write failed.

■ Environmental Variables

CSCI is invoked by `csciServer.sh`, a Transaction Program Name (TPN) in the directory where ACSLS executables reside. This shell script defines CSCI environment variables.

Table 29 shows descriptions of CSCI environment variables for the server subsystem. Following this table is a shell script like the one provided in the distribution of ACSLS/CSCI software. This shell script is started by TCP/IP on the server.

Table 29. CSCI Server Subsystem Environment Variables

Name	Description
CSCI_TRACE	This variable is used to indicate whether CSCI tracing is **ON** or **OFF**.
CSCI_SLEEP_TIME	This variable is used to indicate the PER sleep time at the end of each round-robin loop through the connection table (polling timeout). Default is **100 msec**.
CSCI_INITIAL_CONNECTION	This variable is used to indicate the CSCI's initial connect name to open first. This is the connection service name requiring i/o on initial startup of the CSCI. Clients usually open IPC first; Servers open TCP/IP first.
CSCI_SERVER_NAME	This variable is used to indicate the name of the CSCI Server. It must be set the same for all CSCI's Clients defined to a single CSCI server. **The server name must match its CSCI Clients CSCI_SERVER_NAME.** The matching of the Server and Client CSCI_SERVER_NAME provides an end-to-end logical link between machines. Finally, this name is used later for the "connectionname" in other environment variables. The value assigned to the CSCI_SERVER_NAME must be replicated in the connectionname part of the subsequent variables.
CSCI_connectionname_NET_TYPE	This variable is used to indicate what Network type this CSCI is configured for from the CIF subsystem. The NETTYPE defines the transport layer used by this CSCI. Valid assignments are **LU62** or **IPC**.
CSCI_connectionname_INPUT_ SOCKET	This variable is used to indicate the input socket number to use as input to this CSCI. This number would match the Server or Client application output socket number.

Table 29. CSCI Server Subsystem Environment Variables

Name	Description
CSCI_connectionname_CON_TYPE	This variable is used to indicate this CSCI type. This CON_TYPE defines the CSCI to be either a **SERVER** or a **CLIENT**.
CSCI_connectionname_TRANSLATE_TYPE	This variable is used to indicate this CSCI XDR type to be employed during execution. This can be assigned to either **XDR** or **NONE**.
CSCI_connectionname_DESTINATION_SOCKET	This variable is used to indicate destination or output socket number to be used by the CSCI Server. This socket number would be the matching input socket number to the CSCI Server Application.

Glossary

A

absent cartridge A cartridge that is in the database, but that couldn't be found when all recorded locations for the cartridge were catalogued. If a nonzero retention period is set, the volume status is changed to STATUS_VOLUME_ABSENT.

ACS See Automated Cartridge System.

ACSEL See ACS Event Logger.

ACS Event Logger (ACSEL) The software component that receives messages from other ACSLS components and writes them to an Event Log.

ACS ID A unique identifier for an ACS.

ACSLH See ACS Library Handler.

ACS library A library is composed of one or more ACSs, attached tape drives, and cartridges residing in the ACSs.

ACS Library Handler (ACSLH) The part of the ACSLM that communicates directly with the LMU.

ACSLM See ACS Library Manager.

ACS Library Manager (ACSLM) The software component that validates and routes library requests and responses.

ACSLS See ACS Library Software.

ACSLS database ACSLS database containing information about the location and status of the data or cleaning cartridges. The information includes cell location, scratch status, etc.)

ACSLS platform The server hardware and software that provide the proper environment for ACSLS.

ACS Library Software (ACSLS) Manages ACS library contents and controls ACS library hardware to mount and dismount cartridges on ACS cartridge drives.

ACSLS database A database used by ACSLS to track the library configuration and the locations and IDs of all data or cleaning cartridges in the library.

ACSSA See ACS System Administrator.

ACS System Administrator (ACSSA) The interface between the Command Processor and the rest of the system.

ADI Application Data Interchange.

audit A physical inventory of the contents of all or part of a library.

Automated Cartridge System (ACS) The library subsystem consisting of a single or dual LMU, and 1 to 24 LSMs connected to that LMU.

automated library See library.

beginning of tape (BOT) The location on a tape where written data begins.

B

BOT See Beginning of Tape.

C

CAP See Cartridge Access Port.

CAP ID A unique identifier for the location of a CAP. A CAP ID consists of the ACS ID, the LSM number, and the CAP number.

cartridge A plastic housing containing a length of data recording tape. The tape is threaded automatically when loaded in a transport. A plastic leader block is attached to the tape for automatic threading. The spine of the cartridge can contain an OCR/Bar Code label listing the volume ID.

Cartridge Access Port (CAP) A bidirectional port built into the door panel of an LSM, which provides for the manual entry or automatic ejection of data or cleaning cartridges.

cartridge drive (CD) A device containing two or four cartridge drives and their associated power and pneumatic supplies.

cartridge tape I/O driver Operating system software which issues commands (e.g., read, write, and rewind) to cartridge subsystems.

cartridge transport An electromechanical device that moves tape from a cartridge over a head that writes and reads data from the tape. A transport is distinct from the power and pneumatic sources that supply the electricity and air it needs to function. See cartridge drive.

CCI See client computing system.

CD See cartridge drive.

cell A receptacle in the LSM in which a cartridge is stored.

channel A device that connects the host and main storage with the input and output control units.

client applications Software applications that manage tape cartridge contents. They access tape cartridges by interacting with ACSLS. Any number of client applications can be resident on a client system.

client computing system A computer and an executable image of the operating system.

client software This software manages tape cartridge contents, generates requests for cartridges, and drives data to and from cartridges. The client software is not part of ACSLS.

Client System Component Software which provides an interface between the client computing system's operating system and ACSLS.

Client System Interface (CSI) The software component that translates and routes messages between the ACS Library Manager and the Client System Component.

command access control Limits access to commands.

command area The bottom area of the cmd_proc interface where you enter requests and receive responses.

command processor (cmd_proc) The screen interface of the ACSSA. cmd_proc lets you enter the commands described in Chapter 7.

control path adapter A hardware device which converts a Client Computing System's control protocol to the control protocol of the Sun StorageTek Library Control System.

control unit (CU) A microprocessor-based unit logically situated between a channel and up to sixteen cartridge transports. The CU translates channel commands into transport commands and sends transport status to the channel.

CSE Customer Services Engineer.

CSC Client System Component.

CSI See Client System Interface.

CSI variables Used to define various options to fine-tune communications

between a CSC and the CSI. You change these variables in the acsss_config program.

CU See control unit.

cycle error messages Messages that indicate a library or ACSLS failure.

D

database A collection of interrelated data records. See also ACSLS Database.

data path The network path that allows client applications read/write access to tape cartridges.

data path adapter A hardware device which translates a Client Computing System's data protocol to the data protocol of the Sun StorageTek Control Unit.

display area The top area of the cmd_proc interface that collects messages regarding the status of the library.

dynamic configuration tallows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running.

E

ejected cartridge A cartridge that has been ejected from the library. If a nonzero retention period is set, the cartridge status is changed to STATUS_VOLUME_EJECTED.

end of tape (EOT) The location on a tape where written data ends.

EOT See end of tape.

EPO Emergency Power Off.

EPROM See erasable programmable read only memory.

erasable programmable read-only

memory (EPROM) A special memory chip that can be erased and reprogrammed.

Event Log A file, maintained by the ACSEL, that contains messages describing library and ACSLS events.

Event Logger See ACS Event Logger.

external label identifiers A six-character alphanumeric label on the outside edge of a cartridge used to identify a physical tape cartridge. It may consist of uppercase letters A through Z, numerals 0 through 9, $, #, and blanks.

F

full installation A complete software installation required for new customer sites or for existing sites where a new library has been installed.

H

HLI Host/LMU Interface. One way that the StreamLine Library Manager communicates with a library.

HLI-attached Libraries that are connected to the StreamLine Library Manager through the HLI. These libraries can be connected through a serial interface (serial-attached) or through a TCP/IP interface (TCP/IP-attached).

home location The cell associated with a given cartridge.

I

ID Identifier or identification.

Initial Program Load (IPL) A process that activates a machine reset, initiates wake up diagnostics (from EPROMs) and loads functional code.

inline diagnostics Routines that test components of a subsystem while operating on a time-sharing basis with the functional microcode in the subsystem component.

in-transit cartridges Cartridges between their source and destination locations. Cartridges are considered in-transit if they are in pass-thru ports, robot hands, or playground.

I/O Input/Output.

IPC Interprocess Communication.

IPL See Initial Program Load.

J

journal A sequential log of changes made to the database since the last checkpoint.

L

LAD Lock Access Door.

LAN See local area network.

large CAP (LCAP) A 40-cartridge CAP with the storage cells arranged in four removable magazines of ten cells each. The magazines appear as a single column of 40 cells to the host software.

LCAP See large CAP.

LCU See Library Control Unit.

LED See Light Emitting Diode.

library A library is composed of one or more ACSs, attached tape drives, volumes in the ACSs, and the ACSLS software that controls and manages the ACSs.

library configuration options Allows the customer to specify the number of ACSs in the library and the connections between each ACS and the server system.

library control component Software which controls the mounting and dismounting of cartridges in the ACS.

library control processor Properly configured computer hardware that, with the addition of appropriate software, supports the operation of the Library Control Software.

library control system The library control platform loaded with library control software (ACSLS).

library control software The software components of ACSLS including the library control component, the Client System Interface and Library Utilities.

Library Control Unit The portion of the LSM that controls the picking, mounting, dismounting, and replacing of data abd cleaning cartridges.

library drive A cartridge transport attached to an LSM that is connected to, and controlled by, a client system. Library drives interact with the LCU during automated tape cartridge mount and dismount operations. Library drives interact with a client application during tape data transfer operations. Library drives are individually addressable by the ACSLM and are individually accessible by client applications. See Cartridge Transport.

library errors Errors that occur because the library is offline, has suffered hardware failure, is unavailable, etc.

Library Management Unit (LMU) The portion of an ACS that manages LSM's, allocates their resources, and communicates with ACSLS.

Library Storage Module (LSM) An ACS structure that provides the storage area for cartridges, cartridge drives, CAPs, and the robot necessary for moving them.

light emitting diode (LED) A light emitting device that uses little energy and is used mainly to indicate on/off conditions.

LMU See Library Management Unit.

local area network (LAN) A computer network in which any component in the network can access any other component. This is the type of interface between an LMU and attached LSM's.

LSM See Library Storage Module.

LSM ID A unique identifier for an LSM. The LSM ID consists of the ACS ID and the LSM number.

M

missing cartridge A cartridge that is in the database, but couldn't be found. If a recorded possible location for the cartridge could not be examined due to an offline LSM or a drive not communicating, the cartridge is marked MISSING instead of ABSENT. The cartridge status is changed to STATUS_VOLUME_MISSING.

N

network adapter Equipment that provides an electrical and logical interface between a network and specific attached equipment.

Network Interface (NI) An interface between the server system and the client systems that maintains network connections and controls the exchange of messages. The NI is resident on the server system and each client system.

NI See Network Interface.

O

OCR Optical character recognition.

ONC Open network computing.

Open Systems Interconnection (OSI) A software architecture model of the International Organization for Standardization. The OSI model provides standards for the interconnection of data processing systems.

OSI See Open Systems Interconnection.

OSLAN Open Systems Local Area Network.

P

Pass-Thru Port (PTP) Mechanism that allows a cartridge to be passed from one LSM to another in a multiple LSM ACS.

PCAP See priority CAP.

playground A reserved area of special cells (within an LSM) used for storing diagnostic cartridges and cartridges found in-transit upon power-on and before initialization of the LSM is completed.

pool A collection of tape cartridges having one or more similar features or attributes, such as a pool of scratch tapes.

POST Power-on self-test.

priority CAP (PCAP) A single-cartridge CAP used for priority entry and ejection of cartridges.

processing errors Errors that result from processing or network communication failures.

PROM Programmable read-only memory.

PTP See Pass-Thru Port.

R

RDBMS Relational database management system.

redo log files Backup files used to restore the ACSLS database.

relational database A database that is organized and accessed according to relationships between the data items; relationships are represented by tables.

ROM Read-only memory.

RPC Remote Procedure Call.

S

SCAP See standard CAP.

scratch An attribute of a tape cartridges, indicating that it is blank or contains no useful data.

SCSI Small computer serial interface.

Serial-attached See HLI-attached.

server system The part of the library that is the residence for ACSLS, now referred to as the Library Control System. The Library Control System acts as an interface between a library and client systems.

server system user A person who invokes ACSLS commands, utilities, or procedures on the server system. Server system users are generally site and maintenance personnel (for example, library operators, tape librarians, system administrators, CSEs, and systems personnel).

servo A system that uses feedback to control a process.

silo A commonly used term for an LSM. See Library Storage Module.

SIMM Single inline memory module.

SQL See structured query language.

SRN See service request number.

SSI See Storage Server Interface.

SSR Software Support Representative.

Standard CAP (SCAP) A 21-cartridge CAP with the storage cells arranged in three rows of seven fixed cells.

Storage Server Interface (SSI) A software component, resident on a client system, that translates and routes messages between client applications and the CSI.

structured query language (SQL) A language used to define, access, and update data in a database.

StorageTek Library Console the operator panel software application used for the StreamLine libraries.

system resource variable Used to control the amount of system resources used by ACSLS.

system unit The Library Control Platform.

T

tape library management system (TLMS) A type of client application.

TCP Transmission Control Protocol.

TLMS See tape library management system.

TOD Time of day.

U

UDP User Datagram Protocol.

UNIX An operating system originally developed by Bell Laboratories (now UNIX Systems Laboratories, Inc.) and used by a variety of computer systems.

unsolicited messages Messages that indicate an error or notify you when a particular routine action can be taken.

UOC Usable on codes.

upgrade installation Performed when installing a new version of ACSLS at an existing customer site.

user selectable features and options variables-Used to define various user-selectable features and options.

V

validation errors Errors that result from format and syntax validation performed by cmd_proc.

venter Virtual enter. Entering an unlabeled cartridge with a virtual label.

virtual label A logical volume ID (volser) that can be assigned to a cartridge when its physical label is missing or unreadable.

volser Volume Serial Number.

volume A data or cleaning cartridge.

volume access control Limits access to volumes, usually by the client.

volume identifier A six-character string that uniquely identifies a data or cleaning cartridge to the database.

volume serial number (volser) A synonym for external label identifier.

W

WTM write tape mark.

X

XDR External data representation.

XML Extensible Markup Language. A universal format for structured documents and/or data on the Web.

Index

tuning variables, 33

U

UNIX file backup, 115
unlock command, 275
unscratching volumes, 97
user, show, 270
using a cmd_proc, 4
using the CAP, 81
Utilities
    fix_rc.sh, 151
utilities
    bdb.acsss, 131
    config acs, 135
    config drives, 137
    config lsm, 138
    config port, 139
    db_command, 140
    db_export.sh, 142
    db_import.sh, 143
    del_vol, 143
    drives_media.sh, 145
    ejecting.sh, 149
    find_tty.sh, 150
    fix_rc.sh, 151
    free_cells.sh, 152
    get_license_info, 155
    greplog, 156
    kill.acsss, 157
    moving.sh, 157
    probeFibre.sh, 165
    rc.acsss, 165
    rdb.acsss, 166
    stats_report, 173
    volrpt, 175
    watch_vols, 180

V

vary command, 277
venter command, 284
verifying imported database, 113
    dismounting a volume to verify, 113
    mounting a volume to verify, 113

volrpt utility, 175
volume attributes, 108
volume reports, custom, 122
volume statistics, 36
volume statistics report
    logging volume movement, 124
volume, querying, 257
volumes
    absent, 76, 77
    adding scratch, 94
    delete, 75
    display, 304
    displaying scratch information, 94
    ejecting, 93
    managing scratch, 93
    mounting scratch, 96
    unscratching, 97

W

watch_vols utility, 180